Gaps and requirements for automatic generation of space layouts with optimised energy performance

Du, Tiantian; Turrin, Michela; Jansen, Sabine; van den Dobbelsteen, Andy; Fang, Jian

DOI
10.1016/j.autcon.2020.103132

Publication date
2020

Document Version
Final published version

Published in
Automation in Construction

Citation (APA)
Du, T., Turrin, M., Jansen, S., van den Dobbelsteen, A., & Fang, J. (2020). Gaps and requirements for automatic generation of space layouts with optimised energy performance. *Automation in Construction*, *116*, Article 103132. https://doi.org/10.1016/j.autcon.2020.103132

Automation in Construction 116 (2020) 103132

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Review

Gaps and requirements for automatic generation of space layouts with optimised energy performance

Tiantian Du[a,*], Michela Turrin[a], Sabine Jansen[a], Andy van den Dobbelsteen[a], Jian Fang[b]

[a] Delft University of Technology, Delft, the Netherlands
[b] Computer engineering lab, Delft University of Technology, Delft, the Netherlands

ARTICLE INFO

Keywords:
Space layout design
Automatic generation
Energy performance optimisation

ABSTRACT

Due to the critical need for reducing carbon emissions, the demand for energy-efficient building design is urgent. Studies have shown that space layouts affect energy performance considerably. Energy performance optimisation is able to improve energy performance significantly. However, in order to apply energy performance optimisation to space layouts (EPO), abundant layout alternatives are needed. With the development of computational methods, automatic generation of space layouts (GSL) helps to generate abundant layouts quickly. Therefore, combining GSL with EPO is expected to be greatly helpful for energy-efficient design. This paper investigates 10 relevant studies combining GSL and EPO and analyses their gaps. Furtherly, we extend the analysis to the research on GSL and EPO. 7 GSL methods are categorised and evaluated based on 66 studies, and the requirements for the combination with optimisation are inspected. Regarding EPO, the requirements for energy performance assessment and optimisation are analysed.

1. Introduction

Currently, the energy consumption in buildings constitutes up to 40% of the total primary energy consumption in the U.S and E.U. [1]. Performative computational architecture aims at improving building performance by informing the decisions during the design process based on performance evaluation [2]. It includes the comparison of design alternatives based on quantified performance, and the search for well-performing solutions within large sets of design alternatives. The performative computational architecture has shown great potential in energy saving [3]. Energy performance optimisation is broadly studied, which aims to select the optimal design with minimal energy use.

Space layout design is one of the design tasks taking place in the 'scheme design' and 'design development' stages in the early design phase [4], and one of the most important missions in architectural design. In this paper, the space layout is defined as the allocation of different spaces, and it is decided based on the placement of interior partitions as well as exterior walls. Studies have shown that space layouts can affect building energy performance significantly, regarding heating, cooling, lighting and ventilation demands. A comparison of five space layouts for an office building in the UK was made in [5], and resulted in the biggest difference (difference/the highest demand) of 57% in the heating demand for peak winter and 67% in the lighting demand for peak summer. The same layouts were compared in [6], in which the opening state of windows and interior doors were also changed in addition to the space layout, and resulted in the biggest difference of 65% in the air volume of natural ventilation provided through background vents in peak winter. Three layouts were simulated and compared in [7], which resulted in the biggest difference of 52% in the heating demand for one year and 24% in the cooling demand. Two office layouts in Sweden were simulated and compared in [8], in which window to wall ratio (WWR) was also changed in addition to the space layout, and resulted in the biggest difference of 14% in the heating demand and 57% in the cooling demand. Various layouts for a library building in Turkey were simulated and compared in [9], in which WWR was also changed in addition to the space layout, and resulted in the biggest difference of 19% in the heating demand per day and 20% in the cooling demand per day and 10% in the lighting demand per day. Several layouts for an office building in South Korea were simulated and compared in [10], in which WWR was also changed in addition to

Abbreviations: GSL, automatic generation of space layouts; G-O, automatic generation of space layouts combined with optimisation; EP, energy performance assessment of space layouts; EPO, energy performance optimisation of space layouts; G-EPO, automatic generation of space layouts with optimised energy performance.

* Corresponding author at: Room 250, 1 West, Julianalaan 134, 2628 BL Delft, the Netherlands.
E-mail addresses: T.Du@tudelft.nl (T. Du), M.Turrin@tudelft.nl (M. Turrin), S.C.Jansen@tudelft.nl (S. Jansen), A.A.J.F.vandenDobbelsteen@tudelft.nl (A. van den Dobbelsteen), J.Fang-1@tudelft.nl (J. Fang).

https://doi.org/10.1016/j.autcon.2020.103132
Received 19 June 2019; Received in revised form 2 February 2020; Accepted 14 February 2020
Available online 20 May 2020
0926-5805/

the space layout, and resulted in the biggest difference of 8% in the annual energy use and 15% in the predicted mean vote (PMV). Various layouts for a residential building in Portugal were simulated and compared in [11], in which the window orientation and shading size were also changed in addition to the space layout, and resulted in the biggest difference in thermal discomfort of 33% for the buildings with one floor and 29% for the buildings with two floors.

Evins [12] highlighted the benefits of using computational optimisation during design phases to optimise the energy performance of buildings. His extensive review of precedents in computational energy optimisation reveals a large attention on building envelopes, mechanical systems, and energy generation. Among the analysed precedents, the space layout is rarely used. A similar conclusion can be drawn from the review of Ekici et al. [2], which shows the dominance of energy-related objectives in building optimisation design. The study collected the papers relevant to performative computational architecture including form generation, performance evaluation, and optimisation, with the keywords of 'building design', 'architectural design', 'evolutionary algorithm', 'evolutionary computation', 'swarm intelligence', and 'swarm optimisation'. This review paper shows that WWR, shading, orientation, window dimension, and building shape are the most commonly used design variables during optimisation, among all the form-finding parameters. Among the collected 100 studies, only 6 studies are relevant to space layout design. According to these reviews, it appears that energy performance optimisation has been studied and applied to different design tasks, while it is rarely applied to space layout design. Based on our review, all design tasks for which energy performance optimisation has been applied are represented in parametric variations. However, representing space layouts in parametric variations is difficult. It requires a systematic generation method, and it is not easy to develop when considering the functionality required by a space layout, like non-overlap (two spaces cannot share the same area), non-overflow (spaces cannot go out the layout boundary), and space connections and adjacencies.

1.1. Automatic generation with optimised energy performance

Comparing a large set of alternatives is necessary to identify an optimal design solution. Recent computational development offers an opportunity to automate the generation of design alternatives based on parametric and algorithmic rules. The automatic generation of space layouts (GSL) is to use a computational process to generate a huge set of alternative layouts within a reasonable time span. The automatic generation of space layouts with optimised energy performance (G-EPO), which combines GSL with energy performance optimisation is promising and important for future work, as it can produce a large set of layout alternatives within a reasonable time span, and at the same time, it can compare the building energy performance of these alternatives and search for the optimal designs. Performative computational architecture generally includes three parts: form generation, performance

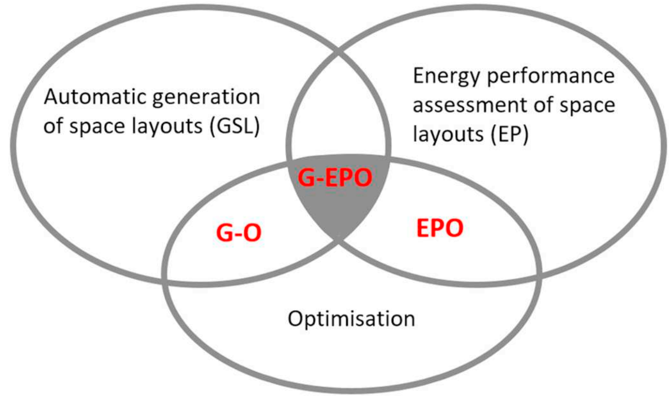


Fig. 2. Relevant research domains of G-EPO.

assessment, and optimisation [2]. Accordingly, G-EPO includes three parts as shown in Fig. 1. The part of GSL regards form-finding and includes algorithmic design, associative geometry and parametric design. The part of performance assessment regards two parts, including layout functionality and energy performance, which are to be maximised or minimised as optimisation objectives. The optimisation part is based on optimisation algorithms and regards the computational process that searches for combinations of design variables which output the layout solutions with the optimal values of the performance indicators. Each part has its specific requirements, and they are also affected by others considering their combination. It is necessary to discuss the gaps and requirements for the combination considering their mutual affects.

1.2. Research questions

The purpose of this paper is to detect the gaps and requirements of G-EPO. As shown in Fig. 2, this topic is relevant to three research domains: GSL, energy performance assessment of space layouts (EP), and optimisation. After our first stage of review, only 12 studies are found focusing on G-EPO. In order to pave the way for future research, we extend the analysis to two relevant research areas, i.e. automatic

Table 1
Keywords used for searching references.

Term (space layout)		Term (energy)		Term (automation)
Space layout Space planning Interior layout Space allocation Floor plan Spatial layout Facility layout	and	Energy use Energy performance Energy saving Green building Sustainable building	and	Automation Optimisation Solution exploration Generation

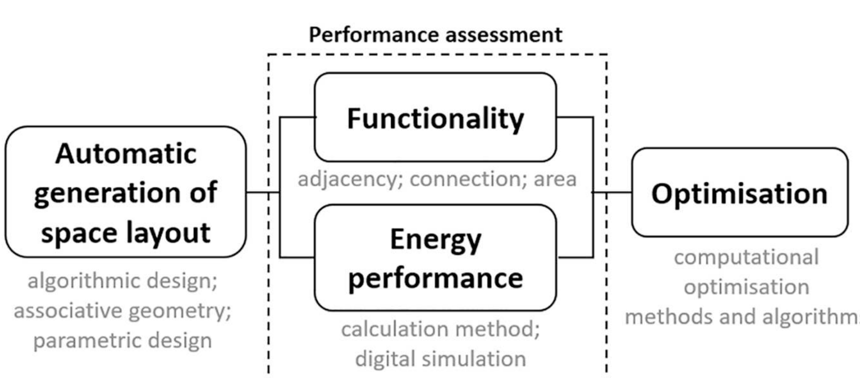


Fig. 1. Framework of G-EPO.

Table 2
Studies relevant to G-EPO.

Ref.	Author	Year	Floor	Pre	Form	Layout representation	Design variables	Objectives/constraints for functionality	Energy calculation methods and/or tools	Objectives for energy performance	Opt	EPI
[21]	Boonstra et al.	2018	Mul	No	Rec	Cell & coordinates	Space location and dimension;	Non-overlap; non-gap;	Toolbox of C++ with Resistor-capacitor-networks	Heating and cooling demands;	/	H + C:10%
[22]	Schwartz et al.	2017	mul	Yes	Rec	Coordinates	Space location and dimension; WWR;	Layout area, number of stories, number of rooms, width and length of each room; adjacency matrix;	jEPLus (a user interface of EneregyPlus)	Annual district heating and cooling consumption;	NSGA-II	/
[23] & [9]	Dino; Dino & Ucoluk;	2017	Mul	Yes	Pol	3D matrix	**Functionality:** space index in 3D location matrix; **Energy performance**: space location; space width-length-height ratio; WWR	Space area; compactness of spaces; regularity: limit space corners; convexity: keep spaces as convex; façade preference of spaces; floor preferences of spaces; spaces adjacency; spaces separation;	OpenStudio	Sum of annual heating, cooling and lighting energy demands; daylighting autonomy;	NSGA-II	H: 23%; C: 25%; L: 11%
[24,25] & [26]	Rodrigues et al.	2014	Sin & Mul	No	Rec	Coordinates	**Functionality:** space size and position; positions of windows and doors; space connectivity; position of floor; **Energy performance**: boundary dimension; boundary orientation; window dimension; space location and dimension; shading dimension;	Connectivity and adjacency; non-overlap; opening orientation; floor dimensions; compactness; non-overflow;	Toolbox of Java integrated with EnergyPlus (acknowledging this by inquiring the author)	Thermal discomfort of different spaces;	SO	33% -29%
[10]	Yi	2016	Sin	Yes	Rec	3D matrix	Boundary dimension; space location and dimension; WWR;	Space connection; non-overlapping;	Ecotect	Annual heating and cooling energy demands; PMV; interior daylight level; interior shading;	SA	H + C: 7.7%; PMV: 13%
[27]	Baušys & Pankrašovaite	2005	Sin	No	Rec	Coordinates	Boundary dimension; space location and dimension; window location and dimension;	Non-overlap; non-overflow; connectivity; minimise space area; minimal natural lighting for some spaces;	Simplified annual calculation	Annual heating cost; annual lighting cost;	GA	/
[28]	Michalek et al.	2002	Sin	Yes	Rec	Coordinates	Space location and dimension; space connectivity; window location and dimension;	Non-overlap; specific path, connectivity, external wall; envelope; wasted space; hall size; access-way size;	Simplified monthly calculation	Annual lighting, heating and cooling cost;	SA & SQP	/
[29]	Caldas	2008	Mul	No	Rec	/	Space dimension; space height; roof tilt and directions; clerestory window under roof; window dimension;	Adjacencies; dimensional constraints; language intensions; explicit and implicit relationships;	DOE 2.1	Energy intensity use (annual energy use and space area)	GA	/
[30]	Sleiman et al.	2017	Mul	No	Rec	Coordinates	Space location; space dimension; location of corridors (space are aligned to corridors);	Floor preference of space; spaces be clustered horizontally or vertically; traveling distance; space area;	TECT within BIM, following EN ISO 16798-1	Energy demand of heating and cooling;	EA	H + C: 8.3%
[31]	Su & Yan	2015	Sin	No	Rec	3D matrix	Locations of patient rooms	Nursing traveling distance;	DIVA (with DAYSIM as engine)	Daylight illuminance;	GA	/

Note: Floor: the floors of generated layouts; Pre: whether the predefined boundary is necessary or not; For: generated space form; Opt: optimisation algorithm; EPI: resulted energy performance improvement; Mul: multi-floors; Sin: single floor; Pol: polygon except for rectangle; Rec: rectangle; NSGA-II: non-dominated sorting genetic algorithm-II; SO: sequential optimisation; SA: simulated annealing; GA: genetic algorithm; SQP: sequential quadratic programming; EA: evolutionary algorithm; PMV: predicted mean vote; WWR: window to wall ratio; '/': not mentioned or not included.

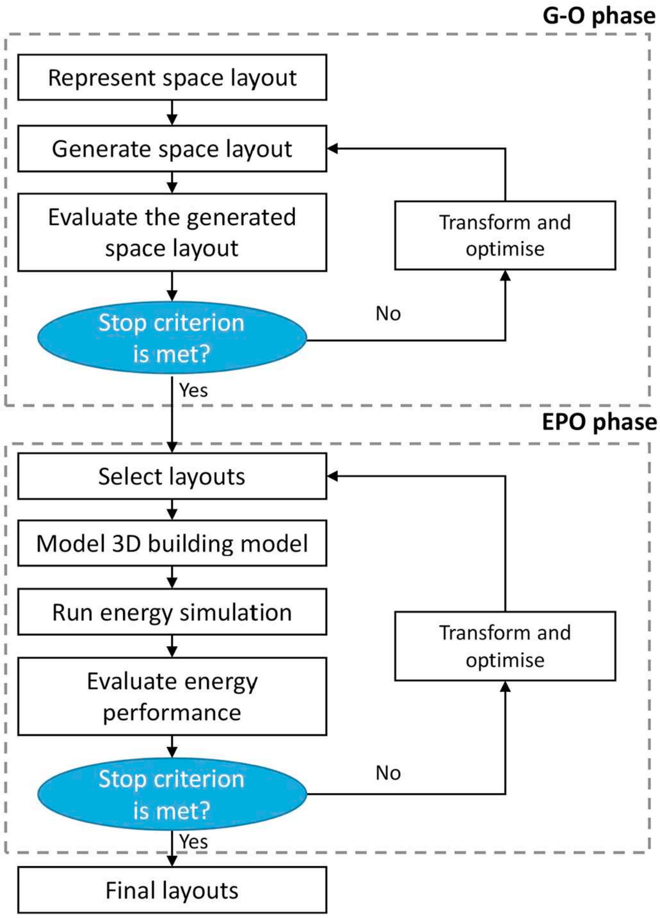


Fig. 3. Workflow of G-EPO, used in [9,21,23–26].

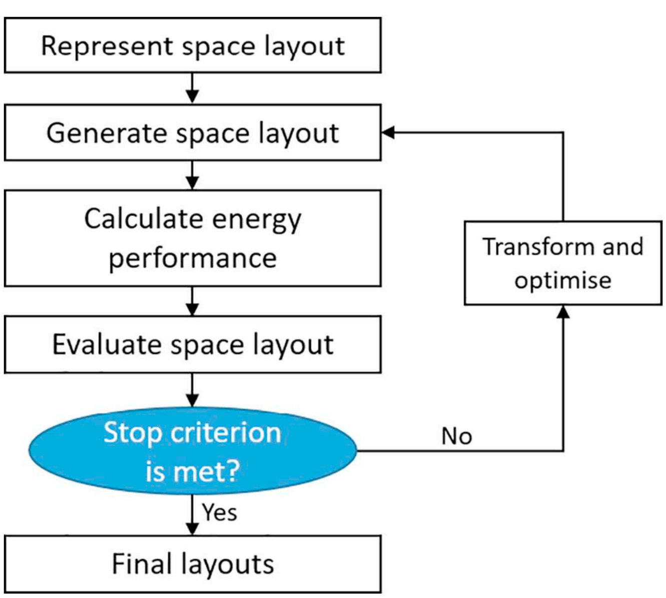


Fig. 4. Workflow of G-EPO, used in [10,22,27–31].

generation of space layouts combined with optimisation (G-O) and energy performance optimisation of space layouts (EPO). The following sub-questions are discussed regarding these two research areas:

- What are the existing GSL methods and what are the criteria for their evaluation?
- What are the requirements for combining GSL methods with optimisation?
- What are the requirements for the energy performance assessment of space layouts?
- What are the requirements for the optimisation of the energy performance of space layouts?

1.3. Selection of references

The keywords used for searching references are shown in Table 1, dividing into space layout, energy, automation. These three terms are used to collect references for Section 2 (G-EPO), and the terms of space layout and automation are used for Section 3 (G-O). The references for Section 2 are also used for Section 4 (EPO). We limit the discipline to architectural design. Although some studies used space layout as the keyword, they actually belong to urban planning like in [13], or neighbourhood planning like in [14]. So we eliminate these studies from the collected references. Another similar concept, facility layout [15], is also easy to be confused with, which has a much wider scope, ranging from the assignment of activities to cites, sites, campus, and buildings [16], to the location of facilities in manufacturing systems [15] and in organisations [17]. In this paper, the studies with the keyword of facility layout which focus on architectural design were selected. Totally, 12 studies are found for G-EPO, and 66 studies are found for GSL.

2. Literature review on G-EPO

We find 12 studies focusing G-EPO. We select 10 of them for detailed review, as the energy indicators used in the other two studies are only relevant to occupant comfort [18,19]. Although some studies changed layout boundary forms and dimensions like in [20], they are not analysed in this review, as their interior space layouts were not changed correspondingly. The review presented herein focuses on the layout generation, energy performance assessment and optimisation. It provides a systematic analysis of the collected references in order to identify the following problems (Table 2):

- the information of the generated layouts: floors of generated layouts, whether the method needs predefined boundary or not, and the generated space form;
- the methods used to represent space layouts (layout representation);
- the design variables meaningful for the layout functionality and/or for the energy performance of the designs;
- the optimisation objectives and constraints for the layout functionality;
- the calculation methods and/or tools for energy performance;
- the optimisation objectives for the energy performance of the designs;
- the optimisation algorithms used for the optimisation process;
- the resulting energy performance improvement (EPI).

2.1. Methodology of G-EPO

There are mainly two methodologies used in these reviewed 10 studies. In the studies of [9,21,23–26], the process of G-EPO is clearly separated into G-O and EPO phases, as shown in Fig. 3. The workflow is as follows:

- G-O: the automatic generation of space layouts combined with optimisation. It includes three steps: first, choosing an appropriate method to represent space layouts; second, fitting the representation of spaces to a suitable generation method and generating the variants of space layouts; third, evaluate the generated space layouts in terms of the requirements of the layout functionality, like adjacency, connection, and area, and deciding whether the stop criterion is

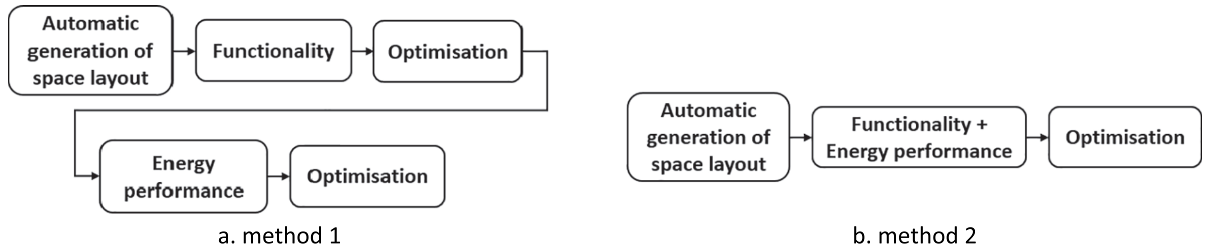


Fig. 5. Two methods of G-EPO.

met. If yes, passing the layout to the next phase; if not, transforming the layout to find a better solution.

- EPO: the energy performance optimisation of space layouts. It includes four steps: first, selecting the appropriate layout from G-O; second, building a 3D building model based on the layout; third, calculating the energy performance with necessary building information, like HVAC system, internal gains, and materials; fourth, evaluating its building energy performance based on the calculation results, and deciding whether the stop criterion is met. If yes, passing the layout as the final layout; if not, transforming the layout to find a better solution. After the iterations of optimisation, the passed layouts are the final layouts.

In the studies of [10,22,27–31], there is not a clear separation between G-O and EPO. Space layouts are generated first; then the energy performance of the generated layouts is calculated; after that, the optimisation algorithm is used to find the optimal layout (Fig. 4). However, with this method, the energy performance of each generated space layout needs to be calculated, resulting in time consuming calculations. Besides, users need to predefine the rough layout at the beginning. So, the generated layouts with this method have narrower variation than with the first method. In the studies of [30,32], the used method following this workflow was called semi-automation.

2.2. Generated layouts and layout representation method

The studies of [21–23,26,29,30] developed the building into multi-floors, while the other studies [10,27,28,31] limited the building to one floor. The studies of [21,24–27,29–31] did not need to predefine a layout boundary, while it was necessary for the others. Most of these studies generated rectangular spaces, while Dino [23] generated polygonal spaces although they were combined rectangles. Two layout representation methods were used to generate layouts: one method used coordinates to represent the location and dimension of spaces [22,24–28,30]; the other method used a 3D matrix to represent spaces and their locations [9,10,23,31]. The study of Boonstra et al. [21] used the combination of the two methods.

2.3. Design variables

Some studies limited the variants within a fixed layout boundary, in which only the design variables relevant to interior space layouts were changed [9,10,23]. The other studies did not limit the change of the boundary, in which the change of space locations and dimensions results in the transformation of boundaries. There is a clear separation between the ***design variables*** for functionality and the ones for energy performance in [9,23–26]. For instance, the space index was only changed for functionality, while WWR was only used for energy performance in [9,23]. The design variables in the collect 10 studies are not uniform: some only used space locations and dimensions [21,30,31], while some also included window dimensions and locations [9,10,22,26–29] and shading dimensions [26].

2.4. Objectives of optimisation

Similar to the design variables, there is also a clear separation between the ***optimisation objectives*** for functionality and energy performance. The objectives for functionality include non-overlap, non-overflow, connectivity and adjacency between spaces, space area, boundary compactness, and traveling distance. The objectives for energy performance include the energy indicators of lighting, heating, cooling, and ventilation, and the comfort indicator of PMV, as well as the daylighting indicators of daylighting autonomy, interior daylight level, and daylight illuminance.

2.5. Energy performance calculation method

Most studies calculated energy performance, except for the study of [31]. In contrast, only several studies [9,10,31] calculated daylighting performance. The tools used for daylighting performance assessment include EnergyPlus, Ecotect and Daysim. Regarding the methods for energy performance calculation, the used methods can be classified into the steady-state calculation method [27,28,30] and dynamic simulation method. Regarding the steady-state calculation method, simplified calculation formulas are used to calculate the energy consumption with empirically determined gain and loss correlation factors, and they are easily to be integrated with the generation of space layouts as well as optimisation. Regarding the dynamic simulation method, the tools used for simulations are capable of the integration with the generation of space layouts and optimisation. For instance, Dino and Ucoluk [9] used OpenStudio and Schwartz et al. [22] used jEPlus (a user interface of EnergPlus) [33], and they customised and extended the tools to couple the parametric simulation with optimisation; Su and Yan [31] used DIVA [34] (a plugin of Grasshopper), and they integrated the plugin with the generation process and the other plugin for optimisation (Galapagos) in Grasshopper. In addition, a toolbox was developed and coded by Rodrigues et al. [26] in JAVA and Boonstra et al. [21] in C++. Rodrigues et al. [26] used an IDF Parser library to edit the IDF file which was further used in EnergyPlus. Boonstra et al. [21] built the resistor-capacitor-network to simulate the thermal building behaviour, then the network was further integrated with the generation and optimisation program.

2.6. Optimisation algorithm

Most studies had multi-objectives, while the studies of [22,29] had one objective. Among the multi-objective studies, most studies converted multi-objectives to a single objective by assigning different weight factors to different objectives [10,26–28,30,31], with which the optimisation results highly depend on the predefined weight factors. Regarding optimisation algorithms, evolutionary algorithms were used in [9,22,27,29–31] and Simulated Annealing was used in [10,28], while the direct search with a sequential optimisation method was used in [26].

Table 3
Analysis of references to elaborate different GSL methods.

GSL method	Ref.	Author	Year	Pre	Floor	Form	Elements to represent space layouts	Actions for generation (actions to change design variables for optimisation)	Constraints and objectives	Opt
Physically based method	[44]	Arvin & House	2002	No	Sin	Rec	Coordinates of centre point of spaces; distance of edges to the centre point; spring to represent the distance;	Adjust attraction and repulsion strength; change the coordinates of spaces and edges;	Space adjacency; space separation; orientation; control shape irregularity by alignment and offset; space area; space area proportion; non-gap;	/
	[45]	Guo & Li	2017	Yes	Mul	Rec	Location of spaces; distance between spaces;	Adjust attraction and repulsion strength; swap space locations; compress building geometry;	Space connections; dimension; shape; building shape;	ES
Mathematical programming method	[24]	Rodrigues et al.	2013	No	Sin	Rec	Size and position of spaces; positions of windows and doors; connectivity between spaces; position of floor plan;	Change space location; rotate along space centre; stretch the space dimensions; mirror opening locations;	Connectivity and adjacency; non-overlap; opening orientation; floor dimensions; compactness; non-overflow;	ES & SHC;
	[46]	Nagy et al.	2017	Yes	Sin	Pol	Edge of spaces; location of spaces; arrangements of desks; location of amenity spaces;	Change space location and dimensions;	Adjacency preference; work style preference; amount and distribution of high-activity zones; sight lines distribution to other desks; daylight amount; unobstructed view to outside;	MOGA
	[47]	Medjdoub & Yannou	2000	No	Mul	Rec	Size and position of spaces; location of windows;	Change space vertex coordinates and space dimensions;	Domain and ratio constraints; space connection; space adjacency; orientation; minimise wasted spaces; non-overflow; non-overlap; non-wasted	EH
Graph-theory aided method	[48]	Verma & Thakur	2010	Yes	Mul	Rec	Index in location matrix; doors location; area; length; width; area to perimeter ratio;	Change the index of spaces in location matrix; vary the wall where doors are located; use Dijkstra's algorithm to find the shortest path; change dimensions of layouts;	Minimise evacuation time;	GA
	[49]	Chatzikonstantinou	2014	No	Mul	Rec	Position of space centre; width and length of spaces;	Change the positon of space centre; use Voronoi to generate rectangle spaces; change the locations of centre points;	Maximise area; minimise cost; maximise proximity or separation between spaces;	NSGA-II
	[19]	Lobos & Trebilcock	2014	Yes	Sin	Rec	Position of nodes;	Change the space locations; apply yED to create final layouts;	Temperature; illumination; percentage of exterior room area; minimise noise;	/
	[50]	Hua	2016	Yes	Sin	Pol	Location of spaces;	Predefine layouts; detect regions from graphs; assign spaces to the regions; change locations of spaces;	Space area; adjacencies between spaces;	SA
	[51]	Nourian et al.	2016	Yes	Sin	Rec	Location of point seeds	Generate adjacency graph; generate a Tutte connectivity graph; use attraction and repulsion force to adjust graph; generate Dual graph; add dimensions to the graph;	/	/
	[52]	Wong & Chan	2009	No	Sin & mul	Rec	Nodes for spaces; edges for adjacencies;	Change the space index in adjacency matrix; transfer matrix to Dual graph; add dimensions to the graph;	Space adjacency; budget; adjacency; relative area ratio of spaces; function deficient;	EA
	[53]	Shekhawat	2015	Yes	Sin	Rec	Location of spaces; adjacencies;	Assign spaces to predefined boundary with spiral-based algorithm; finalize layouts by grouping spaces.	Minimise wasted space; space adjacency;	/
Cell assignment method	[18]	Yi & Yi	2014	Yes	Mul	Rec	Space adjacency; space index in location matrix;	Assign spaces to grids; change space index in location matrix;	Space adjacency; space area; PMV; daylight level; interior/ exterior shading;	SA
	[23]	Dino	2016	Yes	Mul	Pol	Interior partitions; dimensional ratio of spaces;	Assign spaces to voxels; change space locations to avoid overlap and waste space; change space index in the location matrix;	Space area; compactness of spaces; regularity: limit space corners; convexity: keep spaces as convex; façade preference of spaces; floor preferences of spaces; adjacency between spaces; separation between spaces;	EA
	[54]	Yeh	2006	Yes	Mul	Rec	Locations of spaces; weight of adjacencies between spaces;	Assign space to grids; change space index in location matrix;	Site preference; adjacency; space location feasibility;	NN & SA
	[55]	Gero & Kazakov	1998	Yes	Mul	Pol	Location of spaces, defined in genotype;	Assign spaces to layout based on assignment pattern; change values in genotypes;	Minimise travel distances between spaces; minimise travel costs between spaces;	GA
Space splitting method	[56]	Das et al.	2016	Yes	Sin	Rec	Index of spaces in data tree	Split layout based on data tree;	Maximise patient beds; minimise nurse travel distance; maximise connectivity to the existing building; minimise view impedance;	/
	[57]	Koenig & Knecht	2014	Yes	Sin	Rec	Index of spaces in data tree	Split layout based on data tree structure; change layers and values in data tree;	Adjacency between spaces;	ES&GA &GP
Occupant-trace based generation method	[58]	Ghaffarian et al.	2018	Yes	Mul	Irre	Agent's wander rate; separation; cohesion; alignment force; collision;	Generate circulation pattern based on agent trace simulation; assemble negative space as functional space;	Avoid view blocking;	/
Machine learning method	[59]	Huang & Zheng	2018	Yes	Sin	Rec	/	/	/	/
	[60]	Sharma et al.	2017	No	Sin	Rec	/	/	/	/
	[61]	Chaillou	2019	Yes	Mul	Sin+ irre	/	/	/	/

Note: Floor: the floors of generated layouts; Pre: whether the predefined boundary is necessary or not; Form: generated space form; Opt: optimisation algorithm; Mul: multi-floors; Sin: single floor; Pol: polygon except for rectangle; Rec: rectangle; Irre: irregular; ES: evolutionary strategy; SHC: stochastic hill climbing; MOGA: multi-objective genetic algorithm; EH: enumeration heuristic; GA: genetic algorithm; NSGA-II: non-dominated sorting genetic algorithm-II; SA: simulated annealing; EA: evolutionary algorithm; NN: neural networks; GP: genetic programming; '/': not mentioned or not included; '*': the multi objectives are converted to single objective with weighted-sum approach.
Among the actions for generation, the ones used for optimisation are marked in red.
Among all the generation methods, the generation process of machine learning method is different from others, which cannot be divided into representation, generation, and optimisation. So these information is not included for machine learning method in this table.

2.7. Energy performance improvement and conclusions

Based on the results of the 10 studies for detailed review, the highest improvements in the heating, cooling and lighting demand are up to 23% [9], 25% [9] and 11% [9] respectively. This shows that G-EPO is promising to improve building energy performance. Two methods of G-EPO were used (Figs. 3 and 4), and we formulate them as follows: in the first method, functionality is optimised first and then energy performance is optimised (Fig. 5-a); in the second method, functionality and energy performance are optimised as the same time (Fig. 5-b).

However, only 12 studies are found relevant to G-EPO. The limitations of the collected studies are apparent: in the G-O part, only several automatic generation methods were used; in the EPO part, these studies used various energy assessment methods regarding design variables, energy indicators and simulation methods, and they were not uniform. Thus, we review and analyse G-O and EPO separately in Section 3 and Section 4, in order to find solutions to these limitations.

3. GSL method: categorisation and combination with optimisation

Research on GSL started around fifty years ago [35]. There are several review papers on GSL. Helme and Derix [36] collected the projects using GSL in practice; Dutta and Sarthak [37] solely focused on the application of evolutionary computing approaches for space layout design and did not categorise the GSL methods; Nassar [38] discussed the advances in graph theory and analysed their possibilities to be applied to architectural space layout design. The following review studies focus on the methods used for GSL: Frew in 1980 [39] categorised the methods based on whether the boundary was varied or not and how to change space dimensions; Hsu and Krawczyk in 2003 [40] introduced the methods used for space-planning programs separately regarding adjacency, representation, and different actions used among the design process; Lobos and Donath in 2010 [41] collected some relevant studies, but did not categorise the GSL methods; Calixto and Celani in 2015 [42] focused on the used evolutionary algorithms used for GSL. These studies lack the systematic analysis and categorisation of GSL methods. Some of them either only introduced some examples, and some separated the methods either only for representation or only for generation, and the others focused on the evolutionary algorithms. This part of this paper aims at categorising the GSL methods, from the perspective of the generation process of space layouts, considering both representation and generation.

In our previous paper, we classified 4 GSL methods [43]. In this section, 66 studies are found focusing on GSL, and 22 are analysed in detail (Table 3) regarding the information of generated layouts, layout representation, layout generation, constraints and objectives for optimization, and optimization algorithm. We categorise them into 7 GSL methods and explain them in terms of layout representation and generation. After that, these methods are evaluated. As most studies used optimisation for layout functionality, in the last part, the requirements for the combination with optimisation are analysed.

3.1. Design requirements for layout functionality

The design requirements for layout functionality should be satisfied by GSL. Generally, these requirements can be classified into two groups: topological requirements and geometric requirements [24,47], as shown in Table 4. Topological requirements refer to the relative relationship between spaces, including connection, adjacency, and separation between spaces, as well as orientation preferences. Geometric requirements are the ones relevant to dimensional information of spaces and the layout boundary, including width, depth, length, area, and compactness. Additionally, non-overlap and non-overflow should also be satisfied: non-overlap means that two spaces cannot overlap each other; non-overflow means that spaces cannot overflow the layout boundary.

Table 4
Requirement for layout functionality.

Topological requirement	Geometric requirement
Space connection	Width, length and height of space
Space adjacency	Width, length and height of boundary
Space separation	Space area
Orientation preference	Layout area
	Space compactness
	Boundary compactness
	Non-overlap
	Non-overflow

3.2. GSL methods categorisation

Based on the analysis in Tables 3, 7 GSL methods are categorised. These methods are explained as follows.

3.2.1. Physically based method

In this method, space layouts are generated by applying physical forces to the spaces. The layout generation process is reformulated to a process to find the equilibrium between different forces, for instance, the attraction and repulsion in a spring system [44,45]. In this method, a space is represented as a circle or rectangle, and the connection between spaces is represented by the string between circles or rectangles (Fig. 6-a). Regarding the topological resolution, spaces are represented as circles, and attraction and repulsion forces are applied to strings until the equilibrium is reached (Fig. 6-b). Regarding the geometric resolution, space locations are changed by designers, and with this action, the overlaps and gaps between spaces can be removed and the adjacencies and connections between spaces can be changed (Fig. 6-c). Regarding the final layout, users need to manually finalize the layout to satisfy all requirements, like aesthetic intentions (Fig. 6-d). Forces mainly work on space centres, while they can also work on space edges to change the space form [44]. Some plugins in Grasshopper can help to simulate the physical motions, like Kangaroo [62].

3.2.2. Mathematical programming method

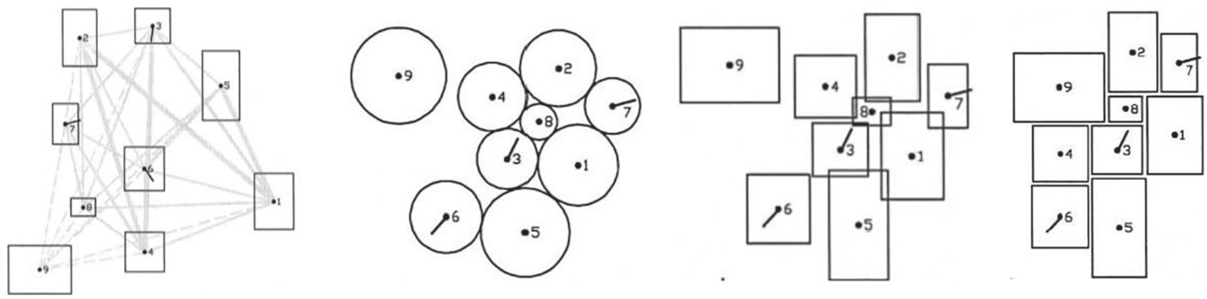
In this method, the design parameters of space layouts and the requirements for layout functionality are transformed into formulas [24,28]. Space locations are represented with the coordinates of centre points, and space connections and adjacencies are controlled by the relative distance between two centre points (Fig. 7-a). The design requirements, like non-overlap and non-overflow, are transformed into constraints, and expressed as mathematical formulas (Fig. 7-b). By changing space locations and dimensions, the feasible layouts are obtained by satisfying all constraints (Fig. 7-c).

3.2.3. Graph-theory aided method

In this method, space adjacencies are transformed to a planar graph, and algorithms for graph theory are used to convert the planar graph into a feasible space layout [51,52]. In this method, the generation process is clearly divided into topology and geometry design. Taking the study of [52] for example: first, the space adjacency preferences are stored in a 2D matrix, which can be varied for alternatives (Fig. 8-a); then, the matrix is transformed to a planar graph, in which nodes represent spaces and links represent connections(Fig. 8-b); algorithms are used to convert the planar graph to a graph which can be converted to a feasible layout, like a dual graph, in which the links can be divided into multi-floors (Fig. 8-c); the final space layout is obtained by inserting geometric information to the graph (Fig. 8-d). Regarding the last step, the geometric information was inserted by designers or architects manually in the study of [52]. Extra steps are needed for the generation of geometric variants, in order to realise the automation for the whole generation process. For instance, in the study of [49], the location of space centre points was used as the starting point, and the middle line between two adjacent points was used as the edge of the rectangle space. After all middle lines were found, the initial floor plan with rectangle spaces was obtained. Then, by changing the locations of the centre point for each space, the width and length are changed correspondingly. In addition to the dual graph used in [51,52], other algorithms can also be used, like Voronoi diagram in [49].

3.2.4. Cell assignment method

In this method, the building geometry is predefined and divided into



a. spaces and strings b. topological resolution c. geometric resolution d. final layout

Fig. 6. Generation process in [44] (also the source of images).

$$R_1(x_1, y_1, w_1, h_1,);$$

$$d_x(R_1, R_2) = max\{V_x(R_1), V_x(R_2)\} - min\{V_x(R_1), V_x(R_2)\} - w(R_1) - w(R_2)$$

$$d_y(R_1, R_2) = max\{V_y(R_1), V_y(R_2)\} - min\{V_y(R_1), V_y(R_2)\} - h(R_1) - h(R_2)$$

a. space location and dimension, and distance between spaces

$$f_2(\mathrm{I}) = \sum_{i=1}^{N_s-1} \sum_{j=i+1}^{N_s} f_{ov}(F_i, F_j) + \sum_{i=1}^{N_s} \sum_{j=1}^{N_a} f_{ov}(F_i, A_j)$$

$$f_{ov}(\mathrm{R}_1, \mathrm{R}_2) = \omega(\mathrm{R}_1 \cap \mathrm{R}_2)\mathrm{h}(\mathrm{R}_1 \cap \mathrm{R}_2)$$

b. formula for non-overlap

c. an example of generated layouts

Fig. 7. Generation process used in [24,25] (also the source of images).

3D cells with the same size. The generation process is reformulated to a process to assign different spaces to the cells [10,23,55,63,64]. First, a matrix is defined by users to represent the cells in the building, and the value in the matrix represents which space is assigned to the corresponding cell (Fig. 9-a); second, spaces are assigned to the cells in the building geometry correspondingly (Fig. 9-b); then, by changing the values in the matrix, the feasible layout can be obtained satisfying both geometric and topological requirements (Fig. 9-c). In addition to using a matrix, a method with a space-filling curve was also used in [18,65], in which spaces were assigned to cells according to the sequence defined in the curve.

3.2.5. Space splitting method

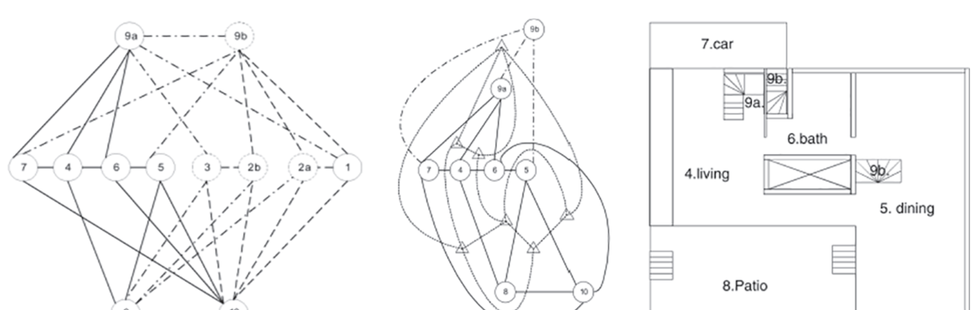
In this method, a predefined floor plan is split recursively following a sequence, which is stored in a data tree [56,57,66]. The node in the data tree represents a space, and the value in the node represents the dimensional information for where the splitting line locates, like the space area. First, a floor plan is defined by users (Fig. 10-a); second, space dimensions and adjacencies are coded into a data tree (Fig. 10-b), which can be varied for layout alternatives; third, the initial layout is recursively split based on the tree data (Fig. 10-c); finally, the final layout is generated after all splits (Fig. 10-d). There are different slicing methods as shown in [56], like slicing by distance, slicing by ratio, and slicing by area.Some splitting strategies can help to generate irregular spaces. For instance, the ice-ray shape grammar was used to generated polygonal spaces in [66], and predefined splitting lines from designers were used to split the layout in [50].

3.2.6. Occupant-trace based method

In this method, a space layout is generated based on occupant tracks, which are obtained by simulating occupant movements [58]. First, occupant movements are simulated, which are controlled by external forces of attraction and repulsion, and affected by the environmental elements, like obstacles and destinations (Fig. 11-a); second, the simulated occupant tracks are used as circulation paths (Fig. 11-b); third, the circulation paths are meshed and converted to feasible spaces

	8	2a	2b	7	5	4	6	3	1	9a	10	9b
8	-	3	3	0	1	3	0	1	0	0	3	0
2a		-	0	0	0	0	0	0	2	0	3	3
2b			-	0	0	0	0	2	0	0	3	3
7				-	0	1	0	0	0	3	3	3
5					-	0	2	0	0	0	2	3
4						-	2	0	0	3	0	0
6							-	0	0	3	1	0
3								-	0	3	3	0
1									-	3	3	3
9a										-	0	3
10											-	0
9b												-

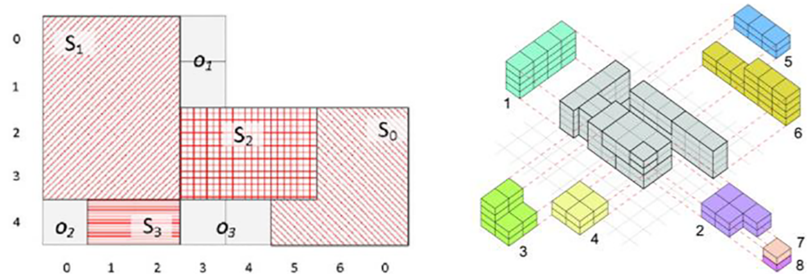
a. adjacency preference matrix



b. planar graph c. dual graph of one floor d. generated space layout of one floor

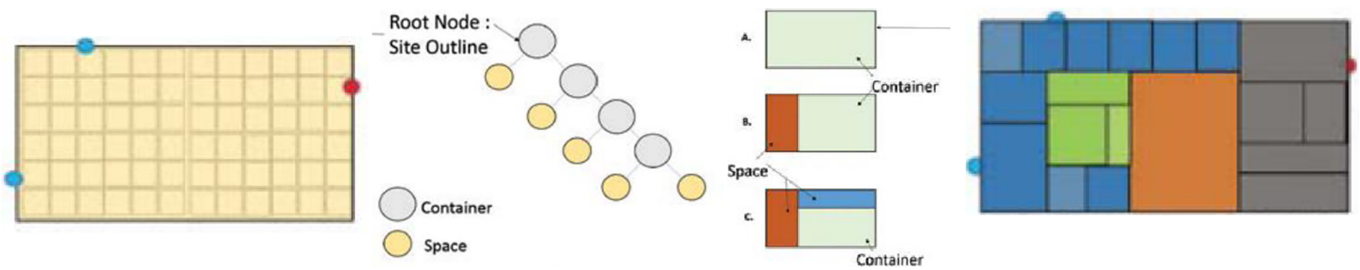
Fig. 8. Generation process used in [52] (also the source of images).

$$A_{space} = \begin{bmatrix} 1 & 1 & 1 & -1 & - & - & - & - \\ 1 & 1,3 & 1,3 & -1 & - & - & - & - \\ 1 & 1,3 & 1,3,2 & 2 & 2 & 2,0 & 0 & 0 \\ 1 & 1,3 & 1,3,2 & 2 & 2 & 2,0 & 0 & 0 \\ -1 & 3 & 3 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$



a. assignment matrix b. spaces are assigned to cells c. generated space layout

Fig. 9. Generation process in [23] (also the source of images).



a. predefined layout b. data tree c. splitting process d. generated space layout

Fig. 10. Generation process in [56] (also the source of images).

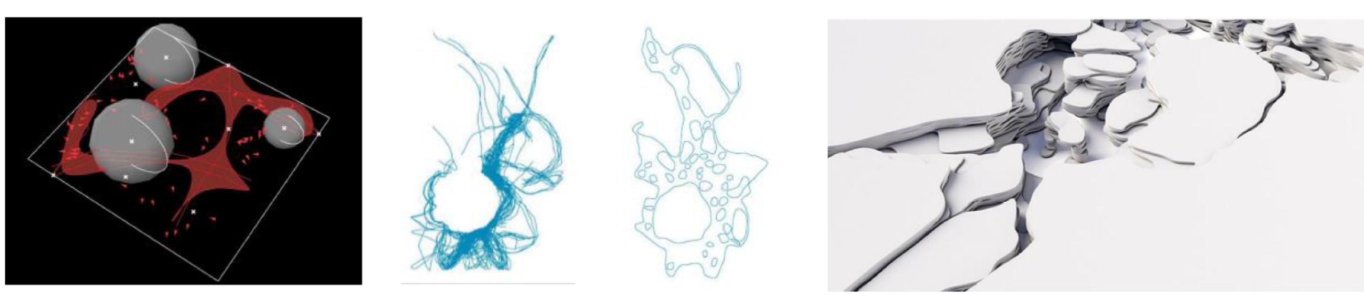
(Fig. 11-c); finally, the left-over spaces are used as the volumes to accommodate functional spaces (Fig. 11-d). Several tools are available to simulate occupant movements, like Quelea in Grasshopper [67] and PEDSIM [68]. This method is broadly used for the site planning [69,70], and some studies used this method to evaluate the existing space layout for renovation [71,72]. A similar concept was applied to the interactive design of the interior space, in which the furniture changed accordingly to occupant movements resulting in different interior spaces [73].

3.2.7. *Machine learning method*

In this method, a model of machine learning is trained based on the dataset with real cases of space layouts, then the trained model is used to generate space layouts with certain inputs [59–61]. The machine learning method is a method to mimic the decision making process of architects based on their expertise and experience [60], without the need to understand thoroughly the logic behind the experience. Taking the study [59] for example, Generative Adversarial Network (GAN) was used for machine learning and the developed method is as follows. First, the real cases of space layouts are collected (Fig. 12-a) and used as dataset. Second, the collected space layouts are labelled manually using different colours to represent spaces, i.e. colour labelled map (Fig. 12-b). Third, one network is trained using the colour labelled maps as input and space layouts as outputs. After this, the model with the network is able to produce the space layouts based on labelled maps. Chaillou [61] furtherly developed this technique into an available tool that can be used by designers, and the design procedure with this tool is as follows: firstly, designers define the layout boundary, the entrance and windows; then the trained model is used to generate the coloured map and add furniture to the coloured map. Regarding the elements used as inputs of dataset, the study of [59] used images of space layouts as inputs, while a natural language description [74] and the features of space layouts (like space adjacency, room area, and layout area) [75] were also used. Additionally, the deep learning approach was also used for the generation of space layouts, which does not need to manually label space layouts for inputs, as shown in [60].

3.2.8. *Classification of relevant studies*

We classify the collected 66 studies based on our categorisation, in Table 5. Among these studies, some combined different GSL methods. The combination takes advantage of the strength of different methods,



a. simulation of occupant movements b. circulation paths c. circulation spaces d. generated space layout

Fig. 11. Generation process in [58] (also the source of images).

a. Example of the collected space layout   b. Corresponding colour labelled map

Fig. 12. Collected space layout and colour labelled map in [59] (also the source of images).

as some methods are easier to generate topological solutions, while others are easier to generate geometric solutions. For instance, Takizawa et al. [76] combined space splitting method and cell assignment method, in which a data tree was used to generate topological solutions and then spaces were assigned to cells accordingly; Guo and Li [45] combined physically based method and cell assignment method, in which physically forces were used for topological solutions and building geometry was optimised within cells.

Table 5
Classify studies into different GSL methods.

GSL method	Authors	Year	Ref.	GSL method	Authors	Year	Ref.
Physically based method	Guo & Li	2017	[45]	Mathematical programming method	Anderson et al.	2018	[77]
	Christensen	2014	[78]		Lee & Ham	2018	[79]
	Biagini et al.	2014	[80]		Nagy et al.	2017	[46]
	Hsu & Krawczyk	2004	[81]		Song et al.	2016	[82]
	Arvin & House	2002	[44]		Boonstra et al.	2016	[83]
	Harada et al.	1995	[84]		Hempel et al.	2015	[32]
	Fortin	1978	[85]		Koenig & Standfest	2014	[86]
Graph-theory aided method	Ślusarczyk	2018	[87]		Suter et al.	2014	[88]
	Wang et al.	2018	[89]		Rodrigues et al.	2013	[24]
	Hua	2016	[50]		Suter	2013	[90]
	Nourian et al.	2016	[51]		Regateiro et al.	2012	[91]
	Shekhawat	2015	[53]		Manthilake	2011	[92]
	Chatzikonstantinou	2014	[49]		Shikder et al.	2010	[93]
	Lobos & Trebilcock	2014	[19]		Ülker & Landa-Silva	2010	[94]
	Verma & Thakur	2010	[48]		Loemker	2006	[95]
	Wong & Chan	2009	[52]		Baušys& Pankrašovaite	2005	[27]
	Schwarz et al.	1994	[96]		Michalek et al.	2002	[28]
	Roth & Hashimshony	1988	[97]		Medjdoub & Yannou	2000	[47]
	Ruch	1978	[98]		Flemming & Chien	1995	[99]
Cell assignment method	Guo & Li	2017	[45]		Cao et al.	1990	[100]
	Blom et al.	2017	[101]		Elshafei	1977	[102]
	Boonstra et al.	2016	[83]	Space splitting method	Das et al.	2016	[56]
	Dino	2016	[23]		Koenig & Knecht	2014	[57]
	Yi	2016	[10]		Takizawa et al.	2014	[76]
	Herr & Ford	2016	[103]		Langenhan et al.	2013	[104]
	Takizawa et al.	2014	[76]		Correia et al.	2012	[66]
	Yi & Yi	2014	[18]		Knecht & Koenig	2010	[105]
	Zawidzki et al.	2011	[106]		Yao et al.	2003	[107]
	Lopes et al.	2010	[108]		Roth et al.	1982	[109]
	Yeh	2006	[54]	Occupant-trace based method	Ghaffarian et al.	2018	[58]
	Gero & Kazakov	1998	[55]		Dzeng et al.	2014	[71]
	Sharpe	1973	[110]		Lee et al.	2012	[72]
Machine learning method	Chaillou	2019	[61]				
	Huang & Zheng	2018	[59]				
	Peng & Zhang	2017	[111]				
	Jain et al.	2015	[74]				
	Merrell et al.	2010	[75]				

3.3. Evaluation of the 7 GSL methods

The 7 methods identified as current possible methods to generate layouts are evaluated on their pros and cons in this subsection, based on a set of criteria.

3.3.1. Criteria to evaluate GSL methods

Four criteria were used to evaluation GSL methods in [57]: performance, reliability, variance and interaction. These criteria are mainly used to evaluate the automation performance. Additionally, we adjust these criteria and add the requirements for space layout design. The criteria are explained as follows:

- Feasibility: whether the generated layouts are feasible or not, considering the requirements for practice.
- User-friendliness: whether the method is easy to be controlled by designers.
- Generation speed: how fast the method can generate layout solutions.

Table 6
Compare the properties of GSL methods.

	Variance			Multi-floor	Capability of Irregularity	Predefined boundary
	Change boundary	Change topology	Change geometry			
Physically based method	Yes	+ + +	+	+	+	No
Mathematical programming method	Yes	+	+ + +	+ +	+ +	No
Graph-theory aided method	Yes	+ + +	+	+ +	+ +	No
Cell assignment method	No	+ +	+ +	+ + +	+ + +	Yes
Space splitting method	No	+ + +	+	+ +	+ +	Yes
Occupant-trace based method	No	/	/	+ +	+ + +	Yes
Machine learning method	Yes	/	/	+ + +	+ + +	No

Note: '/' means that the property cannot be compared. The number of '+' is given based on the method's strength of each property.
'Change boundary' means whether the layout boundary can be changed or not; 'change topology' and 'change geometry' mean the ability of the GSL method to change the topology and geometry of space layouts respectively.

- Variance: how easy the method is used to generate variants.
- Capability of multi-floor: how easy the method is used to generate multi-floors. This is important, as in practice most buildings have multi-floors. This is also an issue for facility layout planning, as shown in [17].
- Capability of irregularity: whether the method can generate an irregular boundary or space, except for rectangle. The more space forms the method can create, the more options designers can have.
- Necessity of predefined boundary: whether the method needs a predefined boundary or not. In practice, the boundary design might happen before or after space layout design, and it can also be the result of interior space layout design. This requires that the GSL method is capable to use a layout boundary predefined by designers, as well as to generate the layout boundary by itself.

3.3.2. Evaluation

The 7 criteria are divided into the ones that can be quantified and the ones that can only be qualified. The ***quantifiable*** criteria include generation speed, variance, capability of multi-floor, capability of irregularity, and necessity of predefined boundary. The ***qualitative*** criteria include feasibility and user-friendliness, and they are the properties that future studies should satisfy. Regarding ***feasibility***, the generated layout should be feasible for practice, considering structure, fire evacuation, construction, and financial cost, etc. Regarding ***user-friendliness***, the representation elements used for the developed method should be suitable for the targeted users. For instance, architects might prefer to use the graphic language, while programmers and engineers might prefer to use numbers.

The quantifiable criteria, except for the generation speed, are compared between the categorised 7 GSL methods in Table 6. Different values are given to different methods according to their strength of each property, marked with '+', except for 'necessity of predefined boundary' and 'change boundary' for variance. The generation speed cannot be compared, as the layouts in different studies have diverse numbers of spaces.

- Variance

The variance cannot be compared based on the total quantity of generated variants in the relevant studies, as they did not show the exact number of variants. The variance can be compared in terms of whether the layout boundary can be changed or not, the ability to change the topology of space layouts, and the ability to change the geometry of space layouts. If the method can change the layout boundary, the variants include the ones with changed boundaries. The process of space layout design with most GSL methods can be divided into the satisfaction of requirements for both topology and geometry. There is a trade-off between the ability to change topology and geometry. For instance, the mathematical programming method is much easier to change geometry with the change of coordination, while extra operators are needed if the topology want to be changed effectively, like rotating, stretching, and mirroring [24]. The graph-theory aided method is much easier to change the topology by changing the index in the adjacency matrix, while in order to change the geometry, extra efforts are needed as explained in Subsection 3.2.3. The cell assignment method is moderate compared to other methods, as the change of index in the assignment matrix with this method causes the change of space adjacencies as well as the dimension of spaces. Occupancy-trace based method and machine learning method cannot be evaluated for 'change typology' and 'change geometry', as they have different generation process from other methods.

- Capability of multi-floor

The capability of multi-floor is compared regarding how easy the method is used to generate multi-floors. So far, most methods have been usable to generate multi-floor layouts [48,49,52,61,112,113], except for the physically based method. But this method can generate multi-floor layouts by combining with other methods, like in [45]. As for the cell assignment method, as long as the predefined cells are multi-floor as well as the corresponding assignment matrix, the generated layout is multi-floor. As for machine learning method, the same model of machine learning can be used to generate the layouts for different floors. Besides, one can envision that as long as the layouts used as the dataset are multi-floor, the generated layouts can be multi-floor. In contrast, the other methods need designers to pre-assign different spaces to different floors.

- Capability of irregularity

The capability of irregularity is purely decided based on the form of generated spaces, as the boundary form can be predefined by designers or it can be the results of combined spaces. The occupant-trace based method generates an organic form which has the highest irregularity [58]. The cell assignment method is easy to generate polygonal spaces with combined cells [10,23,55]. The machine learning method has shown a high capability to generate irregular space forms, as shown in [61], and the space form of generated space layouts is decided based on the space form in the training dataset. However, although some other studies can also generate polygonal spaces, they used the combined method, like mathematical programming method and space splitting method in [46], and graph-theory based method and cell-assignment method in [50]. No study with physically based method is found to generate polygonal spaces.

- Necessity of predefined boundary

A predefined boundary is necessary for cell-assignment method, space splitting method, and occupant-trace based method, while it is not necessary for the others.

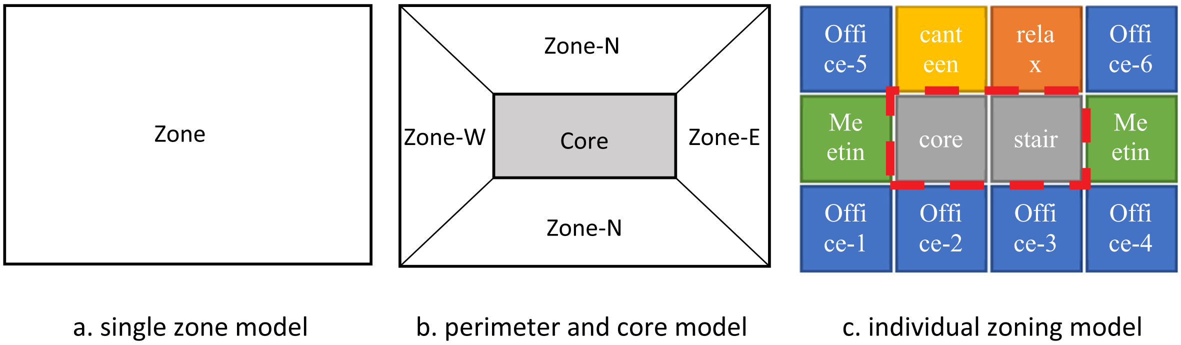


Fig. 13. Three different zoning methods, adapted from [116].

3.4. Optimisation of GSL for layout functionality

We collect the actions taken to change design variables for optimisation, objectives and constraints, and optimisation algorithms of the 22 studies in Table 3. While optimisation algorithms are not discussed as they are not the focus of this paper, the other factors are analysed as follows. Regarding design variables, among the elements used to represent space layouts, only several are used as design variables for optimisation. Especially in graph-theory aided method and space splitting method, only topological design variables are changed, like space indexes in an adjacency matrix and a data tree. Regarding the actions taken to change design variables for optimisation, actions vary with different methods, adaptive to the used design variables. For instance, physically based method changes the force strength, and mathematical programming method alters space coordinates, and graph-theory aided method and cell assignment method vary the space index in the adjacency matrix, and space splitting method adjusts the values in the data tree. Regarding the constraints and objectives, in addition to the ones listed in Table 4, others objectives are also used, like minimal cost [49], minimal evacuation time [48], and maximal view to outside [46]. Besides, some objectives are relevant to the specific building function, like the minimal nurse travel distance in hospital design [56].

4. Requirements for EPO regarding the combination with GSL

EPO includes two parts: EP and optimisation, as shown in Fig. 2. Regarding EP, we detect the requirements for energy indicators and the modelling method for energy performance assessment. Regarding the optimisation part, we analyse the design variables for energy performance optimisation and categorise the methods to reduce computational time.

4.1. Requirements for energy performance assessment

In order to be successfully combined with energy assessments, an GSL method should be useable to calculate a set of meaningful indicators for energy performance and it should allow an appropriate subdivision of the layout into individual thermal zones.

4.1.1. Energy indicators for assessment

Energy performance includes different aspects, i.e. heating, cooling, lighting, and ventilation. In order to fully assess the capability of G-EPO to improve the whole energy performance, all aspects of energy performance (heating, cooling, lighting and ventilation) should be detected. Regarding the assessment boundary, energy indicators can be divided into energy demand, final energy, and primary energy: energy demand is assessed within conditioned building zones, which is calculated based on energy balance; final energy is assessed within the building site, which adds the energy losses from energy distribution systems; primary energy is assessed outside the building site, which adds the energy losses from energy production. The used assessment boundary should be clearly stated in future research. Additionally, only if daylighting and natural ventilation are considered in energy performance assessment, the effect of space layouts on energy performance can be fully identified. So the integration of daylighting and natural ventilation with energy simulation is necessary in energy performance assessment.

4.1.2. Individual zoning method

The simplified steady-state calculation method for energy performance does not need a 3D model as shown in [27,28,30], while the dynamic simulation method needs the 3D thermal zone based model. The modelling process for the dynamic simulation is as follows: first, the model (mostly 2D) obtained from the generation of space layouts is developed into a 3D model by adding height to spaces; then the 3D model is divided into different thermal zones; third, the other information of the building (like HVAC system, internal gains, and materials) is added to the model; finally the dynamic simulation is run with the model.

Different methods of thermal zoning have been used, as shown in [7]: most studies modelled the whole layout as one zone (Fig. 13-a), or separated it into 4 perimeter zones and one core (Fig. 13-b), while some studies separated spaces into individual zones or clustered similar spaces into one zone (Fig. 13-c). The last method is called individual zoning in this paper, and a similar zoning method was proposed in [114]. The first two methods ignore the individual requirements of different spaces and have lower accuracy. Different spaces have various requirements for thermal and visual comfort, as the occupant's activities are different. For instance, as recommended by NEN 16798-1 [115], the heating set-point of offices is 20–25 °C, while the value of corridors is 16–25 °C. By satisfying the individual requirements of different spaces, the whole building energy performance will be drastically decreased, compared to using the same requirements for all spaces. In order to simulate the individual requirements of spaces, the individual zoning method is required.

4.2. Requirements for energy performance optimisation

In order to be successfully combined with the computational parametric optimisation for energy performance, an GSL method should be usable to generate different layout alternatives based on the design variables meaningful for energy performance optimisation. Moreover, EPO should allow a rather fast process to avoid excessive computational time.

4.2.1. Design variables

Computational parametric optimisation is based on the generation and performance assessment of design alternatives. The design alternatives differ from each other based on design variables. When focusing on the energy performance assessment of layouts, the design variables that can affect energy performance depending on the energy balance equation are concluded and classified depending on their relevance with space layout design in Table 7 [117]. The design variables belonging to space layout design can be divided into the design variables within a fixed boundary [5,10,23,28] and the ones within a non-fixed boundary [11,106]. Space properties change with the change of space functions [118–120]. These space properties include: functional requirements, like the set-points for heating, cooling, lighting and ventilation, as well as control types; use of spaces, like the profiles of internal gains resulting from occupancy, appliances and lighting. The envelope design of the building is important for building energy performance, which can influence the impact of space layouts on energy performance [118].

spent on simulations [122]. Surrogate models have been used in different stages of building design, i.e. conceptual design stage [123], sensitivity analysis [124], uncertainty analysis [125], and optimisation [126]. Regarding the design parameters used as inputs, variables of building geometry, windows, and material properties are mainly used [122]. The consumed time is significantly reduced using the surrogate model in comparison to the simulation-based method [127]. For instance, in [128], a surrogate based optimisation method was developed combining ANN and genetic algorithm to help retrofit existing buildings. The results of a case study for a school building show that the total computational time needed for the whole optimisation process involving the training and validation of the ANN model is 3 days. In comparison, the computational time that a simulation based optimisation would be 75 days.

Table 7
Design variables for energy performance optimisation, relating to space layout design.

Design variables of space layouts (with a non-fixed boundary)		Space properties		Envelope design
Space layout design (within a fixed boundary) • Function allocation • Space dimension • Space form • Interior partition • Interior opening	• Boundary dimension • Boundary form • Orientation	Functional requirements • Set-point temperature for heating • Set-point temperature for cooling • Lighting requirements (e.g., illuminance) • Ventilation requirement (e.g., air flow rate) • Control types	Use of spaces • Occupancy, activity and schedule • Internal gains from appliances and lighting • Opening state of windows and doors	• Thermal transmittance • Window area • Window location • Glazing type • Shading type and effectiveness • Air tightness

Note: 'Function allocation' means allocating different functions to different rooms. 'Control types' means the different types of the control for lighting, ventilation, heating and cooling systems. 'Appliances' include the used devices, equipment and machines.

4.2.2. *Reduce computational time*

The building optimisation with multi-objectives is always a time-consuming process. According to Attia et al. [3], the computational time is one of the most important obstacles to the development of energy performance optimisation. Additionally, the energy performance assessment model of space layouts becomes rather complex with individual zoning, which would need more computational time. On the other hand, the detailed dynamic simulation is necessary to obtain the accurate results of energy performance, which makes the energy performance assessment more time consuming. We identify 5 methods to reduce the computational time regarding the elements in energy performance assessment, among which two methods have been used for EPO, i.e. offline simulation [31] and hierarchical structuring of design variables [26].

- Offline simulation

The offline simulation method is to conduct all required simulations before optimisation, in which the rooms with similar situations share the same simulation results. For instance, the rooms facing the same direction share the same daylight illuminance results. In this way, the same simulations do not need to be run for each solution during the optimisation process. The studies using the offline simulation method have shown to be less time consuming [31,121].

- Replacing simulation models with surrogate models

In this method, surrogate models are used to emulate detailed simulation models. The process of surrogate model derivation includes the following steps [122]: first, define the design parameters (inputs) and design objectives (outputs) for the surrogate model; second, create a base building model to generate samples; third, run samples to build database; fourth, fit the surrogate model to the database; fifth, validate the surrogate model for accuracy. The surrogate model is used to predict outcomes instantly based on the given building information, thus saving much computational time which researchers or designers used to

- Sequentially using different assessment methods

There are various methods for energy performance assessment, varying from a simple steady-state calculation to a complex dynamic simulation. Their prediction accuracy and computational time are different. Generally, the computational time is proportional to the prediction accuracy. Correspondingly, optimisation is an iteration process evolving from the preliminary search to the accurate identification of the optimal solution. Invoking the assessment methods from simple to complex in the sequence of optimisation phases can save much time

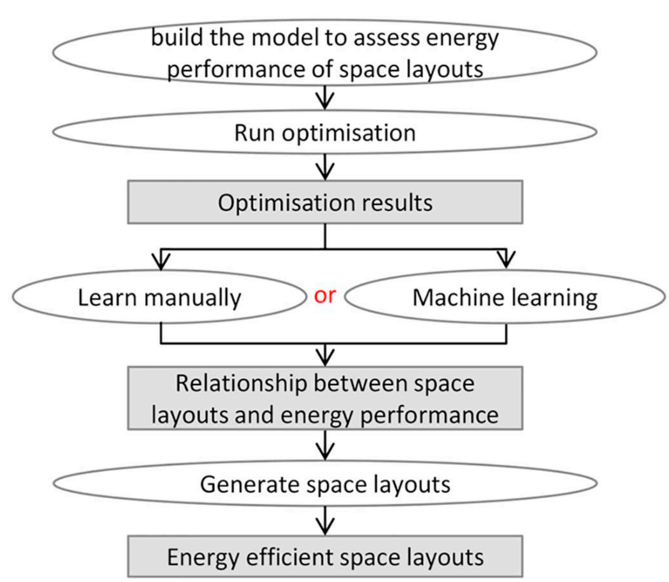


Fig. 14. Proposed alternative methodology for G-EPO.

while keeping similar accuracy, compared to only using complex assessment methods. The study of [129] sequentially used simple yearly calculation, linearized convection calculation, and dynamic simulation following the different phases of optimisation. The results in this study show that the optimisation process using this sequential assessment method saves 2.5 days compared to the method solely using EnergyPlus, which reaches the similar minimum heating and cooling demands.

- Hierarchical structuring of design variables

There are plenty of design variables in building design, and some variables change dependently on others. Structuring hierarchical layers of design variables can avoid infeasible solutions, thus saving the unnecessary computational time, as shown in [130]. The design variables of space layouts are mutually dependent, as shown in Table 7. For instance, the geometry design can be the result of space layout design. Space uses are affected by space layout design. For instance, an open office has a higher occupancy density than a cell office. Thus, structuring hierarchical layers of layout variables can help to avoid infeasible layouts in the generation process. Besides, sequentially invoking different design variables in optimisation process, based on their importance for energy performance, can also help to save computational time, like in Rodrigues et al. [26].

- Hierarchical structuring of optimisation objectives

Similar to design variables, optimisation objectives can also be structured into hierarchical layers. In the study of [131], a target-cascading optimisation method was developed, in which the optimisation objectives were structured into overall performance (overall area and thermal efficiency), thermal comfort, and energy loads. Once the current layer of objectives was satisfied, design variables were passed to the next layer for the optimisation of sub-targets. Regarding EPO, the optimisation objectives can be structured based on space layouts' impact on these objectives. For instance, the study of [116] shows that changing space layouts affects the lighting demand the highest, compared to heating and cooling demands. In this case, the lighting demand can be on the top layer in the objective hierarchy.

5. Conclusions, summaries and recommendations

5.1. Conclusions

In this paper, we collect and review the studies focusing on G-EPO. The review result shows that G-EPO is a promising topic for research and also for architectural design, especially for energy-efficient design. The collected references show promising results, as building energy performance is significantly improved comparing the optimised design with the original design, and the generated layouts are practical and various. Based on this, we extend the analysis to two relevant research domains, i.e. G-O and EPO, in order to find their respective requirements considering their combination. Regarding G–O, 7 GSL methods are categorised based on 66 collected papers. They are evaluated in terms of automation performance and the requirements for space layout design. The requirements for its combination with optimisation are also investigated. Regarding EPO, the requirements for energy performance assessment of space layouts are identified, in terms of energy indicators and zoning method. The design variables for energy performance optimisation are inspected, as well as the methods to reduce computational time.

5.2. Summaries

We summarise the review regarding G-O and EPO.

- G-O phase

Regarding the G-O phase, 7 classified GSL methods are compared regarding their generation speed, feasibility, variance, user-friendliness, capability of multi-floor, capability of irregularity, necessity of predefined boundary. The quantifiable criteria are evaluated and compared between methods in Table 6, which would help designers choose the proper generation method. For instance, if designers have a preference for variance, the mathematical programming method is superior to other methods; if designers prefer to easily generate multi-floors, the cell assignment method is superior to others.

- EPO phase

EPO phase includes energy performance assessment and optimisation. Future research should calculate the different energy indicators (for heating, cooling, lighting and ventilation) as more as possible, as there is a trade-off between different energy indicators. The used assessment boundary should be clearly stated, differing between energy demand, final energy, and primary energy. The integration of daylighting and natural ventilation with energy simulation is also highly recommended for the calculation of energy performance. Regarding the zoning method, the individual zoning method should be used in future research for higher accuracy and modelling the different properties of different spaces. Different properties should be modelled in the assessment model of energy performance for different spaces, like setpoints for heating, cooling, lighting and ventilation, occupancy, and internal loads.

Regarding the optimisation part, future research should develop an effective way to reduce computational time, since it is the predominant obstacle to energy performance optimisation. We identify 5 methods to reduce the computational time. The method of offline simulation needs predefined space layout typologies and massive beforehand simulations, and it is not flexible enough to explore layout variants, but suitable to the designs for a given building type which has specific layout typologies. The method with surrogate models would be an effective way to save computational time for space layouts. It is recommended to test the feasibility of surrogate models for the assessment of energy performance of space layouts. However, as discussed before, space layouts cannot be easily represented in parametric variations, so the choice of design parameters (inputs) and the creation of samples are crucial. The hierarchical methods of design variables, optimisation objectives, and simulation methods do not need predefined layouts. They are more practical and suitable for small-scale design projects.

5.3. Recommendations

We formulate some recommendations regarding the whole process of G-EPO, which would help future research. Generally, there is a trade-off between G-O and EPO. The automatic generation of space layouts is developed from the perspective of designers and its outcomes need high variance and diversity, which requires a fast feedback from EPO. In contrast, in order to have a high accuracy of energy performance, EPO needs detailed models, which is time consuming. Regarding the integration of G-O and EPO, the computational time is the main concern, as well as the compatibility of the used tools with both G-O and EPO. As for the future research, two main methodologies of G-EPO are proposed as follows:

- the current method as shown in Figs. 3–4 with a fast decision process, either with a simplified method (or surrogate model) for energy performance assessment or using a powerful machine to run the process.
- an alternative method (Fig. 14): first, building the parametric optimisation model for energy performance and running optimisation; then, learning the relationship between design variables of space

layouts and energy performance manually or with a machine learning method; finally, integrating the learnt relationship with one of the GSL methods, and the generated layouts are expected to be energy-efficient.

The G-EPO aims to develop the available methodology or tool that can be used by architects and engineers, and release them from the redundant and repeatable work with the computational method. For now, this research area lacks the inputs of the requirements from the possible users, like architects and engineers. It would be helpful to conduct a survey to the possible users for their expectations of G-EPO, for instance, as for architects, the inputs that they prefer to use and the workflow that they would like to follow for space layout design, as for engineers, the outcomes that they expect to obtain from the energy performance optimisation of space layouts.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgements

We would like to show our sincere thanks to Zishun Liu and Liangliang Nan, as they helped us to understand the mathematical formulas used in some references, and Pirouz Nourian, as he helped us to understand the graph-theory aided method. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. We would like to express our gratitude to the reviewers for their constructive feedback and valuable insights.

References

[1] X. Cao, X. Dai, J. Liu, Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade, Energ. Buildings 128 (2016) 198–213, https://doi.org/10.1016/j.enbuild.2016.06.089.
[2] B. Ekici, C. Cubukcuoglu, M. Turrin, I.S. Sariyildiz, Performative computational architecture using swarm and evolutionary optimisation: a review, Build. Environ. 147 (2019) 356–371, https://doi.org/10.1016/j.buildenv.2018.10.023.
[3] S. Attia, M. Hamdy, W. O'Brien, S. Carlucci, Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design, Energ. Buildings 60 (2013) 110–124, https://doi.org/10.1016/j.enbuild.2013.01.016.
[4] Design to construction | AIA ETN, https://www.aiaetn.org/find-an-architect/design-to-construction/ , Accessed date: 7 January 2020.
[5] F. Musau, K. Steemers, Space planning and energy efficiency in office buildings: the role of spatial and temporal diversity, Archit. Sci. Rev. 51 (2008) 133–145, https://doi.org/10.3763/asre.2008.5117.
[6] F. Musau, K. Steemers, Space planning, ventilation and energy efficiency in offices, Int. J. Vent. 8 (2009) 9–22, https://doi.org/10.1080/14733315.2006.11683828.
[7] C.B. de Souza, S. Alsaadani, Thermal zoning in speculative office 07 the connections between space layout and inside temperature control, Proceedings of the 1st Building Simulation and Optimization Conference, Loughborough, UK, 2012, pp. 417–424 http://www.ibpsa.org/proceedings/BSO2012/6C1.pdf , Accessed date: 2 February 2020.
[8] H. Poirazis, Å. Blomsterberg, M. Wall, Energy simulations for glazed office buildings in Sweden, Energ. Buildings 40 (2008) 1161–1170, https://doi.org/10.1016/j.enbuild.2007.10.011.
[9] I.G. Dino, G. Üçoluk, Multiobjective design optimization of building space layout, energy, and daylighting performance, J. Comput. Civ. Eng. 31 (2017) 04017025, , https://doi.org/10.1061/(ASCE)CP.1943-5487.0000669.
[10] H. Yi, User-driven automation for optimal thermal-zone layout during space programming phases, Archit. Sci. Rev. 59 (2016) 279–306, https://doi.org/10.1080/00038628.2015.1021747.
[11] E. Rodrigues, A.R. Gaspar, Á. Gomes, Automated approach for design generation and thermal assessment of alternative floor plans, Energ. Buildings 81 (2014) 170–181, https://doi.org/10.1016/j.enbuild.2014.06.016.
[12] R. Evins, A review of computational optimisation methods applied to sustainable building design, Renew. Sust. Energ. Rev. 22 (2013) 230–245, https://doi.org/10.1016/j.rser.2013.02.004.
[13] C.A. Conceicao Antonio, J.B. Monteiro, C.F. Afonso, Optimal topology of urban buildings for maximization of annual solar irradiation availability using a genetic algorithm, Appl. Therm. Eng. 73 (2014) 422–435, https://doi.org/10.1016/j.applthermaleng.2014.08.007.
[14] A. Karatas, K. El-Rayes, Optimizing tradeoffs among housing sustainability objectives, Autom. Constr. 53 (2013) 83–94, https://doi.org/10.1016/j.autcon.2015.02.010.
[15] A. Drira, H. Pierreval, S. Hajri-Gabouj, Facility layout problems: a survey, Annu. Rev. Control. 31 (2007) 255–267, https://doi.org/10.1016/j.arcontrol.2007.04.001.
[16] R.S. Liggett, Automated facilities layout: past, present and future, Autom. Constr. 9 (2000) 197–215, https://doi.org/10.1016/S0926-5805(99)00005-9.
[17] A. Ahmadi, M.S. Pishvaee, M.R. Akbari Jokar, A survey on multi-floor facility layout problems, Comput. Ind. Eng. 107 (2017) 158–170, https://doi.org/10.1016/j.cie.2017.03.015.
[18] H. Yi, Y.K. Yi, Performance based architectural design optimization: Automated 3d space layout using simulated annealing, Proceedings of ASHRAE/IBPSA—Building Simulation Conference, Atlanta, GA, USA, 2014, pp. 292–299 https://experts.illinois.edu/en/publications/performance-based-architectural-design-optimization-automated-3d- , Accessed date: 2 February 2020.
[19] D. Lobos, M. Trebilcock, Building performance information and graphs approach for the design of floor plans, Arquiteturarevista 10 (2014) 23–30, https://doi.org/10.4013/arq.2014.101.03.
[20] P. McKeen, A. Fung, The effect of building aspect ratio on energy efficiency: a case study for multi-unit residential buildings in Canada, Buildings 4 (2014) 336–354, https://doi.org/10.3390/buildings4030336.
[21] S. Boonstra, K. van der Blom, H. Hofmeyer, M.T.M. Emmerich, J. van Schijndel, P. de Wilde, Toolbox for super-structured and super-structure free multi-disciplinary building spatial design optimisation, Adv. Eng. Inform. 36 (2018) 86–100, https://doi.org/10.1016/j.aei.2018.01.003.
[22] Y. Schwartz, R. Raslan, I. Korolija, D. Mumovic, Integrated building performance optimisation: coupling parametric thermal simulation optimisation and generative spatial design programming introduction and aims, Proceedings of the 15th International Building Performance Simulation Association, San Francisco, 2017, pp. 1222–1229, , https://doi.org/10.26868/25222708.2017.316.
[23] I.G. Dino, An evolutionary approach for 3D architectural space layout design exploration, Autom. Constr. 69 (2016) 131–150, https://doi.org/10.1016/j.autcon.2016.05.020.
[24] E. Rodrigues, A.R. Gaspar, Á. Gomes, An evolutionary strategy enhanced with a local search technique for the space allocation problem in architecture, part 1: methodology, Comput. Aided Des. 45 (2013) 887–897, https://doi.org/10.1016/j.cad.2013.01.001.
[25] E. Rodrigues, A.R. Gaspar, Á. Gomes, An evolutionary strategy enhanced with a local search technique for the space allocation problem in architecture, part 2: validation and performance tests, Comput. Aided Des. 45 (2013) 898–910, https://doi.org/10.1016/j.cad.2013.01.003.
[26] E. Rodrigues, A.R. Gaspar, Á. Gomes, Improving thermal performance of automatically generated floor plans using a geometric variable sequential optimization procedure, Appl. Energy 132 (2014) 200–215, https://doi.org/10.1016/j.apenergy.2014.06.068.
[27] R. Baušys, I. Pankrašovaite, Optimization of architectural layout by the improved genetic algorithm, Civil Engineering and Management 13 (2005) 37–41, https://doi.org/10.1080/13923730.2005.9636328.
[28] J. Michalek, R. Choudary, P. Papalambros, Architectural layout design optimization, Eng. Optim. 34 (2002) 37–41, https://doi.org/10.1080/03052150214016.
[29] L. Caldas, Generation of energy-efficient architecture solutions applying GENE_ARCH: an evolution-based generative design system, Adv. Eng. Inform. 22 (2008) 59–70, https://doi.org/10.1016/j.aei.2007.08.012.
[30] H.A. Sleiman, S. Hempel, R. Traversari, S. Bruinenberg, An assisted workflow for the early design of nearly zero emission healthcare buildings, Energies 10 (2017) 993, https://doi.org/10.3390/en10070993.
[31] Z. Su, W. Yan, A fast genetic algorithm for solving architectural design optimization problems, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 29 (2015) 457–469, https://doi.org/10.1017/S089006041500044X.
[32] S. Hempel, J. Benner, K.-H. Hafele, Generating early design alternatives based on formalized requirements and geospatial data, Proceedings of the 32nd CIB W78 Conference, Eindhoven, the Netherlands, 2015, pp. 255–264 http://itc.scix.net/pdfs/w78-2015-paper-026.pdf , Accessed date: 2 February 2020.
[33] JEPlus-an EnergyPlus simulation manager for parametrics, http://www.jeplus.org/wiki/doku.php?id=start , Accessed date: 2 February 2020.
[34] J. Niemasz, DIVA for rhino, http://diva4rhino.com/ , Accessed date: 2 February 2020.
[35] P.H. Levin, Use of Graphs to Decide the Optimal Layout of Buildings, Architect 14, (1964), pp. 809–815 https://discovery.nationalarchives.gov.uk/details/r/C10976838 , Accessed date: 2 February 2020.
[36] L. Helme, C. Derix, Spatial configuration: semi-automatic methods for layout generation in practice, The Journal of Space Syntax 5 (2014) 35–49 http://128.40.150.106/joss/index.php/joss/article/view/201/pdf , Accessed date: 14 June 2019.
[37] K. Dutta, S. Sarthak, Architectural space planning using evolutionary computing approaches: a review, Artif. Intell. Rev. 36 (2011) 311–321, https://doi.org/10.

1007/s10462-011-9217-y.
[38] K. Nassar, New advances in the automated architectural space plan layout problem, Proceedings of the International Conference on Computing in Civil and Building Engineering, Nottinham, the UK, 2010, p. 6 https://pdfs.semanticscholar.org/5ebe/2e80f6e0c94df293ad95f33ea00622aece0a.pdf , Accessed date: 2 February 2020.
[39] R.S. Frew, A survey of space allocation algorithms in use in architectural design in the past twenty years, Proceedings of the 17th Design Automation Conference, IEEE, Minneapolis, MN, USA, 1980, pp. 165–174, , https://doi.org/10.1109/DAC.1980.1585243.
[40] Y.C. Hsu, R.J. Krawczyk, New generation of computer aided design in space planning methods: a survey and a proposal, Proceedings of the 8th International Conference on Computer Aided Architectural Design Research in Asia, Bangkok, Thailand, 2003, pp. 101–115 http://papers.cumincad.org/cgi-bin/works/paper/caadria2003_b1-3 , Accessed date: 2 February 2020.
[41] D. Lobos, D. Donath, The problem of space layout in architecture: a survey and reflections, Arquitetura Revista 6 (2010) 136–161, https://doi.org/10.4013/arq.2010.62.05.
[42] V. Calixto, G. Celani, A literature review for space planning optimization using an evolutionary algorithm approach: 1992-2014, Proceedings of the 19th Conference of the Iberoamerican Society of Digital Graphics, Florianópolis, SC, Brasil, 2015, pp. 662–671, , https://doi.org/10.5151/despro-sigradi2015-110166.
[43] T. Du, M. Turrin, S. Jansen, A. Van Den Dobbelsteen, N. Biloria, A review on automatic generation of architectural space layouts with energy performance optimization, Proceedings of the 4th International Conference on Building Energy, Environment, Melbourne, Australia, 2018, pp. 856–861 http://www.cobee2018.net/assets/pdf/p/283.pdf , Accessed date: 16 January 2020.
[44] S.A. Arvin, D.H. House, Modeling architectural design objectives in physically based space planning, Autom. Constr. 11 (2002) 213–225, https://doi.org/10.1016/S0926-5805(00)00099-6.
[45] Z. Guo, B. Li, Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system, Frontiers of Architectural Research 6 (2017) 53–62, https://doi.org/10.1016/j.foar.2016.11.003.
[46] D. Nagy, D. Lau, J. Locke, J. Stoddart, L. Villaggi, R. Wang, D. Zhao, D. Benjamin, Project discover: an application of generative design for architectural space planning, Proceedings of the 17th Symposium on Simulation for Architecture and Urban Design, Toronto, Canada, Society for Computer Simulation International, 2017, pp. 59–66 https://dl.acm.org/doi/10.5555/3289787.3289794.
[47] B. Medjdoub, B. Yannou, Separating topology and geometry in space planning, Comput. Aided Des. 32 (2000) 39–61, https://doi.org/10.1016/S0010-4485(99)00084-6.
[48] M. Verma, M.K. Thakur, Architectural Space Planning Using Genetic Algorithms, in: Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE), IEEE, Singapore, 2010, pp. 268–275, https://doi.org/10.1109/ICCAE.2010.5451497.
[49] I. Chatzikonstantinou, A 3-dimensional architectural layout generation procedure for optimization applications: DC-RVD, Proceedings of the 2014 ECAADe Conference, Northumbria, UK, 2014, pp. 287–296 http://cumincad.scix.net/cgi-bin/works/Show?_id=ecaade2014_163&sort=DEFAULT&search=ioannis&hits=7 , Accessed date: 2 February 2020.
[50] H. Hua, Irregular architectural layout synthesis with graphical inputs, Autom. Constr. 72 (2016) 388–396, https://doi.org/10.1016/j.autcon.2016.09.009.
[51] P. Nourian, S. Rezvani, S. Sariyildiz, A syntactic architectural design methodology: integrating real-time space syntax analysis in a configurative architectural design process, Proceedings of the 9th International Space Syntax Symposium, Seoul, South Korea, 2013, pp. 048:1–048:15 http://resolver.tudelft.nl/uuid:6edc5ae1-bfa4-4542-86e8-013e4edcfa0d , Accessed date: 2 February 2020.
[52] S.S.Y. Wong, K.C.C. Chan, EvoArch: an evolutionary algorithm for architectural layout design, Comput. Aided Des. 41 (2009) 649–667, https://doi.org/10.1016/j.cad.2009.04.005.
[53] K. Shekhawat, Automated space allocation using mathematical techniques, Ain Shams Engineering Journal 6 (2015) 795–802, https://doi.org/10.1016/j.asej.2015.02.008.
[54] I.C. Yeh, Architectural layout optimization using annealed neural network, Autom. Constr. 15 (2006) 531–539, https://doi.org/10.1016/j.autcon.2005.07.002.
[55] J.S. Gero, V.A. Kazakov, Evolving design genes in sapce layout planning problems, Artif. Intell. Eng. 12 (1998) 163–176, https://doi.org/10.1016/S0954-1810(97)00022-8.
[56] S. Das, C. Day, A. Hauck, J. Haymaker, D. Davis, Space plan generator: rapid generation & evaluation of floor plan design options to inform decision making, Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture, Ann Arbor, 2016, pp. 106–115 http://papers.cumincad.org/cgi-bin/works/Show?acadia16_106 , Accessed date: 2 February 2020.
[57] R. Koenig, K. Knecht, Comparing two evolutionary algorithm based methods for layout generation: dense packing versus subdivision, Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 28 2014, pp. 285–299, , https://doi.org/10.1017/S0890060414000237.
[58] M. Ghaffarian, R. Fallah, C. Jacob, Organic architectural spatial design driven by agent-based crowd simulation, Proceedings of the 18th Symposium on Simulation for Architecture and Urban Design, San Diego, CA, United States, 2018, pp. 173–180 https://dl.acm.org/doi/10.5555/3289750.3289767.
[59] W. Huang, H. Zheng, Architectural drawings recognition and generation through machine learning, Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, Mexico City, 2018, pp. 156–165 http://cumincad.scix.net/data/works/att/acadia18_156.pdf , Accessed date: 2 February 2020.
[60] D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, DANIEL: A deep architecture for automatic analysis and retrieval of building floor plans, Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition, Kyoto, Japan, 2017, pp. 420–425, , https://doi.org/10.1109/ICDAR.2017.76.
[61] S. Chaillou, AI + Architecture Towards New Approach, master Thesis Graduate School of Design, Harvard University, 2019, https://doi.org/10.9783/9781949057027-006.
[62] D. Piker, Kangaroo - Grasshopper, http://www.grasshopper3d.com/group/kangaroo? , Accessed date: 2 February 2020.
[63] A. Menges, Biomimetic design processes in architecture: morphogenetic and evolutionary computational design, Bioinspiration and Biomimetics 7 (2012) 15003, , https://doi.org/10.1088/1748-3182/7/1/015003.
[64] J.H. Jo, J.S. Gero, Space layout planning using an evolutionary approach, Artif. Intell. Eng. 12 (1998) 149–162, https://doi.org/10.1016/S0954-1810(97)00037-X.
[65] Y.K. Yi, Dynamic coupling between a Kriging-based daylight model and building energy model, Energ. Buildings 128 (2016) 798–808, https://doi.org/10.1016/j.enbuild.2016.05.081.
[66] R. Correia, J.P. Duarte, A. Leitao, GRAMATICA: A general 3D shape grammar interpreter targeting the mass customization of housing, Proceedings of the 30th ECAADe Conference, Prague, Czech Republic, 2012, pp. 489–496 http://papers.cumincad.org/data/works/att/ecaade2012_273.content.pdf , Accessed date: 2 February 2020.
[67] A.J. Fischer, Quelea: Agent-based Design for Grasshopper Agent-Based Design for Grasshopper, https://www.food4rhino.com/app/quelea-agent-based-design-grasshopper , Accessed date: 2 February 2020.
[68] C. Gloor, PEDSIM - Pedestrian Crowd Simulation, http://pedsim.silmaril.org/ , Accessed date: 2 February 2020.
[69] N. Popov, Generative urban design with cellular automata and agent based Modelling, Proceedings of the 44th Annual Conference of the Architectural Science Association, Auckland, New Zealand, 2010 https://www.academia.edu/1434082/Generative_urban_design_with_cellular_automata_and_agent_based_modelling , Accessed date: 2 February 2020.
[70] A. Turner, A. Penn, Encoding natural movement as an agent-based system: an investigation into human pedestrian behaviour in the built environment, Environment and Planning B: Planning and Design 29 (2002) 473–490, https://doi.org/10.1068/b12850.
[71] R.J. Dzeng, C.W. Lin, F.Y. Hsiao, Application of RFID tracking to the optimization of function-space assignment in buildings, Autom. Constr. 40 (2014) 68–83, https://doi.org/10.1016/j.autcon.2013.12.011.
[72] H.Y. Lee, I.T. Yang, Y.C. Lin, Laying out the occupant flows in public buildings for operating efficiency, Build. Environ. 51 (2012) 231–242, https://doi.org/10.1016/j.buildenv.2011.11.005.
[73] J.R. Chang, Architectural Engineering and Technology, Delft University of Technology (Ed.), HyperCell:A Bio-inspired Design Framework for Real-time Interactive Architectures, 2018, https://doi.org/10.7480/abe.2018.1 Ph.D. Thesis.
[74] M. Jain, A. Sanyal, S. Goyal, C. Chattopadhyay, G. Bhatnagar, Automatic rendering of building floor plan images from textual descriptions in English, Journal of Latex Class Files 14 (2015) 1–8 http://arxiv.org/abs/1811.11938 , Accessed date: 2 February 2020.
[75] P. Merrell, E. Schkufza, V. Koltun, Computer-generated residential building layouts, ACM Transactions on Graphics (TOG) 29 (2010) 1, https://doi.org/10.1145/1866158.1866203.
[76] A. Takizawa, Y. Miyata, N. Katoh, Enumeration of floor plans based on a zero-suppressed binary decision diagram, Proceedings of the 19th International Conference on Computer- Aided Architectural Design Research in Asia, Kyoto, Japan, 2014, pp. 25–44, , https://doi.org/10.1260/1478-0771.13.1.25.
[77] C. Anderson, C. Bailey, A. Heumann, D. Davis, Augmented space planning: using procedural generation to automate desk layouts, Int. J. Archit. Comput. 16 (2018) 164–177, https://doi.org/10.1177/1478077118778586.
[78] J.T. Christensen, The generation of possible space layouts, Proceedings of the 32nd ECAADe Conference, Newcastle upon Tyne, England, 2014, pp. 239–246 http://papers.cumincad.org/data/works/att/ecaade2014_100.content.pdf , Accessed date: 2 February 2020.
[79] C. Lee, S. Ham, Automated system for form layout to increase the proportion of standard forms and improve work efficiency, Autom. Constr. 87 (2018) 273–286, https://doi.org/10.1016/j.autcon.2017.12.028.
[80] C. Biagini, V. Donato, D. Pellis, Preliminary design through graphs: a tool for automatic layout distribution, ICONARP International Journal of Architecture and Planning 2, 2014, pp. 1–13 http://iconarp.selcuk.edu.tr/iconarp/article/view/60 , Accessed date: 2 February 2020.
[81] Y. Hsu, R.J. Krawczyk, Form development with spatial character, Proceedings of the 22nd ECAADe Conference, Copenhagen, Denmark, 2004, pp. 1–8 http://papers.cumincad.org/cgi-bin/works/Show?2004_278 , Accessed date: 2 February 2020.
[82] H. Song, J. Ghaboussi, T.-H. Kwon, Architectural design of apartment buildings using the implicit redundant representation genetic algorithm, Autom. Constr. 72 (2016) 166–173, https://doi.org/10.1016/j.autcon.2016.09.001.
[83] S. Boonstra, K. van der Blom, H. Hofmeyer, M.T.M. Emmerich, Super-structure and super-structure free design search space representations for a building spatial design in multi-disciplinary building optimisation, Electronic Proceedings of the 23rd International Workshop of the European Group for Intelligent Computing in Engineering, Krakow, Poland, 2016, pp. 1–10 https://dspace.lboro.ac.uk/dspace-jspui/handle/2134/8337 , Accessed date: 2 February 2020.
[84] M. Harada, A. Witkin, D. Baraff, Interactive physically-based manipulation of

discrete/continuous models, Proceedings of the SIGGRAPH, Los Angeles, USA, 1995, pp. 199–208 http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/Web/People/aw/pdf/discrete.pdf , Accessed date: 2 February 2020.
[85] G. Fortin, BUBBLE: relationship diagrams using iterative vector approximation, Proceedings of the 15th Conference on Design Automation, Las Vegas, Nevada, 1978, pp. 145–151 https://dl.acm.org/doi/10.5555/800095.803079.
[86] R. Koenig, M. Standfest, Evolutionary multi-criteria optimization for building layout planning: Exemplary application based on the PSSA framework, Proceedings of the 32nd eCAADe Conference, Newcastle upon Tyne, England, 2014, pp. 567–574, , https://doi.org/10.13140/2.1.4951.4888.
[87] G. Ślusarczyk, Graph-based representation of design properties in creating building floorplans, Comput. Aided Des. 95 (2018) 24–39, https://doi.org/10.1016/j.cad.2017.09.004.
[88] G. Suter, F. Petrushevski, M. Šipetić, Operations on network-based space layouts for modeling multiple space views of buildings, Adv. Eng. Inform. 28 (2014) 395–411, https://doi.org/10.1016/j.aei.2014.06.004.
[89] X.Y. Wang, Y. Yang, K. Zhang, Customization and generation of floor plans based on graph transformations, Autom. Constr. 94 (2018) 405–416, https://doi.org/10.1016/j.autcon.2018.07.017.
[90] G. Suter, Structure and spatial consistency of network-based space layouts for building and product design, Comput. Aided Des. 45 (2013) 1108–1127, https://doi.org/10.1016/j.cad.2013.04.004.
[91] F. Regateiro, J. Bento, J. Dias, Floor plan design using block algebra and constraint satisfaction, Adv. Eng. Inform. 26 (2012) 361–382, https://doi.org/10.1016/j.aei.2012.01.002.
[92] I.D. Manthilake, Evolutionary Building Layout Optimisation, Ph.D. Thesis Department of Civil and Building Engineering, Loughborough University, 2011, https://dspace.lboro.ac.uk/dspace-jspui/handle/2134/8337 , Accessed date: 2 February 2020.
[93] S. Shikder, A. Price, M. Mourshed, Interactive constraint-based space layout planning, Proceedings of 18th CIB World Building Congress, Salford, UK, 2010, pp. 112–122 http://www.irbnet.de/daten/iconda/CIB_DC24120.pdf , Accessed date: 2 February 2020.
[94] Ö. Ülker, D. Landa-Silva, A 0/1 integer programming model for the office space allocation problem, Electron Notes Discrete Math. 36 (2010) 575–582, https://doi.org/10.1016/j.endm.2010.05.073.
[95] T.M. Loemker, Designing with machines: solving architectural layout planning problems by the use of a constraint programming language and scheduling algorithms, Proceeding of the 2nd International Conference of the Arab Society for Computer Aided Architectural Design, Sharjah, United Arab Emirates, 2006, pp. 88–106 http://papers.cumincad.org/data/works/att/ascaad2006_paper7.content.pdf , Accessed date: 2 February 2020.
[96] A. Schwarz, D.M. Berry, E. Shaviv, On the use of the automated building design system, Comput. Aided Des. 26 (1994) 747–762, https://doi.org/10.1016/0010-4485(94)90013-2.
[97] J. Roth, R. Hashimshony, Algorithms in graph theory and their use for solving problems in architectural design, Comput. Aided Des. 20 (1988) 373–381, https://doi.org/10.1016/0010-4485(88)90214-X.
[98] J. Ruch, Interactive space layout: A graph theoretical approach, Proceedings of the 15th Design Automation Conference, Las Vegas, Nevada, 1978, pp. 152–157, , https://doi.org/10.1109/DAC.1978.1585162.
[99] U. Flemming, S. Chien, Schematic layout design in SEED environment, J. Archit. Eng. 1 (1995) 162–169, https://doi.org/10.1061/(ASCE)1076-0431(1995)1:4(162).
[100] X. Cao, Z. He, Y. Pan, Automated design of house-floor layout with distributed planning, Comput. Aided Des. 22 (1990) 213–222, https://doi.org/10.1016/0010-4485(90)90050-M.
[101] K. Van Der Blom, S. Boonstra, H. Hofmeyer, M. Emmerich, A super-structure based optimisation approach for building spatial designs, Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering-ECCOMAS VII, Crete Island, Greece, 2017, pp. 3409–3422, , https://doi.org/10.7712/100016.2044.10063.
[102] A.N. Elshafei, Hospital layout as a quadratic assignment problem, J. Oper. Res. Soc. 28 (1977) 167–179, https://doi.org/10.1057/jors.1977.29.
[103] C.M. Herr, R.C. Ford, Cellular automata in architectural design: from generic systems to specific design tools, Autom. Constr. 72 (2016) 39–45, https://doi.org/10.1016/j.autcon.2016.07.005.
[104] C. Langenhan, M. Weber, M. Liwicki, F. Petzold, A. Dengel, Graph-based retrieval of building information models for supporting the early design stages, Adv. Eng. Inform. 27 (2013) 413–426, https://doi.org/10.1016/j.aei.2013.04.005.
[105] K. Knecht, R. Koenig, Generating floor plan layouts with k–d trees and evolutionary algorithms, Proceedings of the GA2010-13th Generative Art Conference, Milano, Italy, 2010, pp. 238–253 https://www.generativeart.com/on/cic/GA2010/2010_18.pdf , Accessed date: 2 February 2020.
[106] M.H. Zawidzki, K. Tateyama, I. Nishikawa, The constraints satisfaction problem approach in the design of an architectural functional layout, Eng. Optim. 43 (2011) 943–966, https://doi.org/10.1080/0305215X.2010.527005.
[107] B. Yao, H. Chen, C.-K. Cheng, R. Graham, Floorplan representations, ACM Trans. Des. Autom. Electron. Syst. 8 (2003) 55–80, https://doi.org/10.1145/606603.606607.
[108] R. Lopes, T. Tutenel, R.M. Smelik, K.J. De Kraker, R. Bidarra, T. Hague, A constrained growth method for procedural floor plan generation, Proceedings of the 11th International Conference on Intelligent Games and Simulation, Leicester, United Kingdom, 2010, pp. 13–23 https://graphics.tudelft.nl/~rafa/myPapers/bidarra.GAMEON10.pdf , Accessed date: 2 February 2020.
[109] J. Roth, R. Hashimshony, A. Wachman, Turning a graph into a rectangular floor plan, Build. Environ. 17 (1982) 163–173, https://doi.org/10.1016/0360-1323(82)90037-3.
[110] R. Sharpe, Optimum space allocation within buildings, Build. Sci. 8 (1973) 201–205, https://doi.org/10.1016/0007-3628(73)90001-7.
[111] W. Peng, F. Zhang, Machines' perception of space: employing 3D Isovist methods and a convolutional neural network in architectural space classification, Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, Cambridge MA, 2017, pp. 474–481 http://papers.cumincad.org/cgi-bin/works/paper/acadia17_474 , Accessed date: 2 February 2020.
[112] E. Rodrigues, A.R. Gaspar, Á. Gomes, An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique, Autom. Constr. 35 (2013) 482–498, https://doi.org/10.1016/j.autcon.2013.06.005.
[113] K. Matsuzaki, T. Irohara, K. Yoshimoto, Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization, Comput. Ind. Eng. 36 (1999) 487–502, https://doi.org/10.1016/S0360-8352(99)00144-8.
[114] P. Raftery, M. Keane, J. O'Donnell, Calibrating whole building energy models: an evidence-based methodology, Energ. Buildings 43 (2011) 2356–2364, https://doi.org/10.1016/j.enbuild.2011.05.020.
[115] Netherlands Standardization Institute, NEN-16798-1 Energy Performance of Buildings - Part 1: Indoor Environmental Input Parameters for Design and Assessment of Energy Performance of Buildings Addressing Indoor Air Quality, Thermal Environment, Lighting and Acoustics - Module M1–6, https://connect.nen.nl/Standard/Detail/207600?compId=10037&collectionId=0, (2015) , Accessed date: 2 February 2020.
[116] T. Du, S. Jansen, M. Turrin, A. Van Den Dobbelsteen, Impact of space layout on energy performance of office buildings coupling daylight with thermal simulation, Proceedings of the 13th REHVA World Congress CLIMA, Burcharest, Romania, 2019, pp. 1–8, , https://doi.org/10.1051/e3sconf/201911103077.
[117] T. Du, S. Jansen, M. Turrin, A. van den Dobbelsteen, Effects of Architectural Space Layouts on EnergyPerformance: A Review, Sustainability, 12 (2020) 1829, https://doi.org/10.3390/su12051829.
[118] Netherlands Standardization Institute, NEN-EN-ISO 13790 energy performance of buildings-calculation of energy use for space heating and cooling, https://connect.nen.nl/Standard/Detail/119188?compId=10037&collectionId=0, (2008) , Accessed date: 2 February 2020.
[119] Netherlands Standardization Institute, NEN-EN 15193–1: Energy Performance of Buildings -Energy Requirements for Lighting -Part 1: Specifications, Module M9, https://connect.nen.nl/Standard/Detail/3526912?compId=10037&collectionId=0, (2017) , Accessed date: 2 February 2020.
[120] Netherlands Standardization Institute, NEN-EN 16798–3: Energy Performance of Buildings-Ventilation for Buildings- Part 3: Ventilation for Non-residential Buildings. Performance Requirements for Ventilation and Room-Conditioning Systems, https://connect.nen.nl/Standard/Detail/3531933?compId=10037&collectionId=0, (2017) , Accessed date: 2 February 2020.
[121] B. Coffey, Using Building Simulation and Optimization to Calulate Lookup Tables for Control, Ph.D. Thesis Architecture - Building Science, University of California, Berkeley, 2011https://escholarship.org/content/qt1202p562/qt1202p562.pdf , Accessed date: 2 February 2020.
[122] P. Westermann, R. Evins, Surrogate modelling for sustainable building design – a review, Energ. Buildings 198 (2019) 170–186, https://doi.org/10.1016/j.enbuild.2019.05.057.
[123] R.E. Edwards, J. New, L.E. Parker, B. Cui, J. Dong, Constructing large scale surrogate models from big data and artificial intelligence, Appl. Energy 202 (2017) 685–699, https://doi.org/10.1016/j.apenergy.2017.05.155.
[124] W. Tian, R. Choudhary, G. Augenbroe, S.H. Lee, Importance analysis and meta-model construction with correlated variables in evaluation of thermal performance of campus buildings, Build. Environ. 92 (2015) 61–74, https://doi.org/10.1016/j.buildenv.2015.04.021.
[125] Y.J. Kim, Comparative study of surrogate models for uncertainty quantification of building energy model: Gaussian process emulator vs. polynomial chaos expansion, Energ. Buildings 133 (2016) 46–58, https://doi.org/10.1016/j.enbuild.2016.09.032.
[126] T. Wortmann, Genetic evolution vs. function approximation: benchmarking algorithms for architectural design optimization, Journal of Computational Design and Engineering 6 (2019) 414–428, https://doi.org/10.1016/j.jcde.2018.09.001.
[127] A. Prada, A. Gasparella, P. Baggio, On the performance of meta-models in building design optimization, Appl. Energy 225 (2018) 814–826, https://doi.org/10.1016/j.apenergy.2018.04.129.
[128] E. Asadi, M.G. da Silva, C.H. Antunes, L. Dias, L. Glicksman, Multi-objective optimization for building retrofit: a model using genetic algorithm and artificial neural network and an application, Energ. Buildings 81 (2014) 444–456, https://doi.org/10.1016/j.enbuild.2014.06.009.
[129] A.P. Ramallo-González, D.A. Coley, Using self-adaptive optimisation methods to perform sequential optimisation for low-energy building design, Energ. Buildings 81 (2014) 18–29, https://doi.org/10.1016/j.enbuild.2014.05.037.
[130] W. Wang, H. Rivard, R. Zmeureanu, An object-oriented framework for simulation-based green building design optimization with genetic algorithms, Adv. Eng. Inform. 19 (2005) 5–23, https://doi.org/10.1016/j.aei.2005.03.002.
[131] R. Choudhary, A. Malkawi, P.Y. Papalambros, A hierarchical design optimization framework for building performance, Proceedings of the 8th International IBPSA Conference, Eindhoven, Netherlands, 2003, pp. 179–186 http://www.ibpsa.org/proceedings/BS2003/BS03_0179_186.pdf , Accessed date: 2 February 2020.